

开源时代

OPEN SOURCE TIMES

2009年7月刊 总第十期

技术新知:

GCC开发计划及其分支介绍
从WinCE看dijyos的生存空间和价值
Linux-Netfilter&iptables
实现机制的分析及应用

开源业界:

IDC: 2008年中国Linux市场收入达2300万美元
Novell公司Linux业务仍然处于亏损状态

社区扫描:

红帽Fedora11: 颠覆Linux难用说
关于Linux/Windows的若干谎言

行业观察:

开源软件的是与非
FOSS品牌塑造需要克服的七点错误



freeBSD®

专家专栏:

FreeBSD安全小组成员delphij专访

FreeBSD是一份性能出众的UNIX操作系统, 支持众多的硬件平台。在国内鲜有他的开发人员。今天我们有幸请到他们当中的一位——delphij, 他活跃于FreeBSD项目相关mailist、ChinaUnix和newsmth的FreeBSD板块。作为国内资深FreeBSD committer, 让他谈谈目前的FreeBSD项目的发展现状, 以及国内用户和开发人员如何参与到类似于FreeBSD项目的开源社区中来。

本期推荐:《开源时代》总第10期有奖问卷调查





2009年7月刊 总第十期

编辑出品:  ChinaUnix.net

网络发行: ChinaUnix

主编: 江晖
技术主编: 樊强
执行编辑: 周荣茂
内容编辑: 周荣茂 覃里
唐川 周平
技术编委: 高延斌 马路遥
白金
美术编辑: 林在子
交流论坛: bbs.chinaunix.net
杂志下载: www.chinaunix.net
www.itpub.net
www.ixpub.net

联系我们:
rmzhou@staff.chinaunix.net
投稿邮箱:
rmzhou@staff.chinaunix.net

媒体支持:  IT168.com

广告联系: 温玉琴
电话: 010-82658790
手机: 13801339139
E-mail: wyq@it168.com

内容目录

卷首语

004

开源业界

IDC: 2008年中国Linux市场收入达2300万美元	005
Novell公司Linux业务仍然处于亏损状态	006
IBM高管称Vista利于Linux部署	007
Dell称大部分用户并不需要最新的Linux版本	008
甲骨文CEO: 未来几年将大量投资Java	008
高通重定义上网本采用Linux待机超8小时	009
英特尔收购Wind River意在Linux市场	010
Novell广告直指红帽Linux 国内Linux市场竞争白热化?	011
戴尔为中小企业预置开源软件包	011
Google在I/O大会上展示Android2.0	012
中小企业如何选择Linux备份工具?	013
欧盟欲通过扩大保障软件消费者法律 恐分裂开源界	013
RHCE十周年: 38000个红帽认证工程师遍布全球	014
IDC: 开源企业应用软件正在兴起	015

社区扫描

Fedora11发布 全面支持ext4文件系统	016
SourceForge开始接管Ohloh网站	016
Eclipse用户中使用Linux的数量赶超Windows	017
关于Linux/Windows的若干谎言	017
多数超级计算机使用Linux操作系统	019
红帽Fedora11: 颠覆Linux难用说	020
开源社区的认可来自于贡献	021
Linux之父: 2.6.30内核将带来“新世界秩序”	022
革命到底: Linux系统何时走向完美?	023

行业观察

开源软件的是与非	025
FOSS品牌塑造需要克服的七点错误	028
开源非“快速致富之匙”需耐心探寻商业模式	030
盗版软件：不值得冒险	031
CIO，是时候在企业中部署开源软件了	033
MIPS会改变Android命运吗？	036

专家专栏

专注于服务器操作系统的FreeBSD是如何炼成的	038
--------------------------	-----

本期推荐

ChinaUnix《开源时代》读者综合调查问卷	043
-------------------------	-----

技术新知

MySQL Replication优化的一些个人总结	044
软RAID部署系统分区之恢复攻略	046
VMware 环境下OpenBSD 4.5以及Gnome 2.24的安装	065
Linux集群系统Vmware虚拟机建设方案	069
Linux-Netfilter&iptables实现机制的分析及应用	091
从WinCE看dijos的生存空间和价值	100
GCC开发计划及其分支介绍	102

版权声明

杂志内容来自ChinaUnix社区及互联网，电子杂志的宗旨是为了更好地传递开源最新自寻和技术经验。如有版权问题敬请联系，我们将会第一时间做出处理。

致谢

本杂志得到ChinaUnix网站Linux时代社区版主的大力支持，技术文章大部分来自版主推荐，更多技术文章可以访问Linux时代精华区。
本刊分析评论部分文章来自IT168技术频道。

网友热评

109

卷首语

十个月前，ChinaUnix 尝试着推出了《开源时代》电子杂志，如今，第十期，2009 年 7 月刊也已经推出。我们朝着国内最好的 Linux 和开源软件技术的开放、自由、分享和协作平台又迈进了一步。在过去的十个月中，我们努力着做好每一期杂志，通过新栏目设置、板块的调整、排版和内容生产方式的改进，我们在杂志的制作方面积累了经验；通过业界观察和技术文章的约稿，我们积累了一批开源业界的技术撰稿人；通过业界专家访谈和技术沙龙的报道，我们积累了一批资深的开源业界专家和讲师。更难得可贵的是，通过这十期杂志内容的积累，我们有单期超过 15 万份的杂志下载量，我们积累了大量的读者，这是我们杂志在十个月来最大的收获，这也是我们前进和发展的动力。

对于目前的《开源时代》杂志来说，要想上一个新台阶，必须要在内容上下功夫，而不仅仅是现在的内容架构和比例。我们必须需要加入新内容，以及新的内容发掘和生产方式。也许十期并没有一百期来的重要，但这十个月来，我们和读者朋友们也是辛苦地在摸索，在为杂志定位，在为杂志的确定发展方向。我们的目标是做好下一个十期，只有这样，我们才有可能期待第一百期。

近日，国内开源业界也是热闹非凡，6 月 29 日至 30 日，一年一度的开源中国开源世界高峰论坛在北京举办。会议通过大会报告研讨和圆桌会议多向互动，主题涉及移动互联网和开放终端，Linux 和开源软件应用和发展，以及云计算、绿色开源等 IT 领域的前沿技术。作为本次大会的合作方，ChinaUnix 也积极地参与到大会的组织和新闻报道中去。在这次大会中，ChinaUnix 就最近一些开源业界的问题和动向，分别采访了 Linux 基金会主席 Jim Zemlin、Ubuntu 创始人 Mark Shuttleworth 和 OpenOffice.org 项目负责人 Louis Suarez-Potts 等人。有关本次大会的内容以及视频采访内容请访问本次大会的 ChinaUnix 官方网站。

另外，不得不提的就是 chinaUnix 自己的技术沙龙，过去的一个月里，ChinaUnix 技术沙龙在北京、西安和成都分别举行了三场，内容各有不同，北京的技术沙龙侧重虚拟化和集群技术，西安和成都的则是网络性能优化和高可用。还有就是即将在 8 月 28 日举办的中国系统架构师大会，这将是我们在以往沙龙基础上的一次整合和飞跃，通过将近一年来对技术沙龙组织和运作的熟悉，期待我们的大会能够取得成功，大会官网已经上线，更多的大会进展和会务动向，请关注 ChinaUnix 网站的更新。

我相信，在中国，Linux 和开源软件将会得到更好的发展。在过去的十个月中，我们看到了 Android 和 Moblin 等开源项目的崛起，在将来的十个月中，我们也许能够看到 Androd 和 Moblin 能够在移动终端，甚至桌面 Linux 取得更大的突破。也许在《开源时代》百期来临之际，Linux 和开源软件已经深入人心，不再是阳春白雪，《开源时代》也许得出日报，名为《今日开源》。

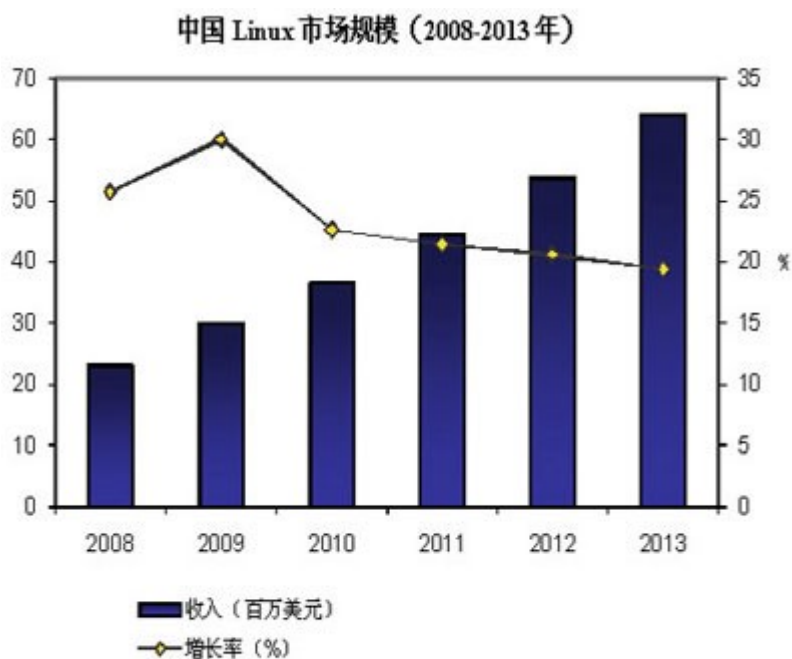
周荣茂

开源业界

IDC：2008 年中国 Linux 市场收入达 2300 万美元

6月24日消息，据 IDC 最新发布的报告《中国 Linux 市场 2009-2013 年预测与分析》显示，2008 年中国 Linux 总体市场收入达到 2300 万美元，增长率为 26%。随着技术的成熟，对于用户来说，Linux 作为一种操作系统软件，无论是性能还是稳定性都日臻完善。Linux 厂商也在市场推广上不遗余力，使中国市场对于 Linux 的接受与部署程度不断增加。

IDC 中国软件与服务研究部高级分析师刘宁蔚认为：“2009 年是中国 Linux 市场发展的关键年，经济危机使 Linux 获得更多市场机会。如果厂商能抓住机遇，增强客户对 Linux 的兴趣和信心，那么 Linux 将会进入一个崭新的发展阶段。” IDC 预测，在 2009 至 2013 年中，中国 Linux 操作系统总体市场的收入将会以 23% 的五年复合增长率保持高速增长。



数据来源：IDC 中国 Linux 市场 2009-2013 年预测与分析

2008 年的中国 Linux 市场可谓有失亦有得。虽然市场在一定程度上受到了下滑的宏观经济以及雪灾、地震等自然灾害的影响，失去了一些大项目的机会；但是也出现了很多新的增长点，包括厂商加强与 SI/ISV 合作、行业应用推广、Linux 虚拟化、上网本等。

IDC 对 2008 年中国 Linux 市场的重点发现如下：

从 LinuxSOE（服务器操作系统）的行业分布来看，2008 年，电信/媒体、金融和政府行业继续成为对 LinuxSOE 收入贡献最大的 3 个行业，总共带来 78.8% 的 SOE 市场收入。值得关注的是，随着国家经济振兴政策的出台，一些目前带来收入较少的行业比如医疗、交通和中小企业，将具有巨大的发展潜力。

政府正版化项目的减少使得 LinuxCOE（客户端操作系统）增长受到了一定影响，因此 LinuxCOE 厂商在不断寻找新的增长点。首先，一些定制化的 LinuxCOE 产品如老人 PC、农村 PC

在 2008 年启动的 PC 下乡项目中可望觅得发展良机。其次，上网本的风行也给 LinuxCOE 市场带来了希望。这些新兴 COE 产品的平均价格都远远高于传统形式的 COEOEM 价格，对 LinuxCOE 市场收入的增长将起到关键推动作用。

本土 Linux 厂商纷纷积极参与了 2008 年底中国政府出台的“核高基”重大扶持项目并存在极大的中标可能性。核高基项目对申请厂商的资质和未来产业推广计划做出了非常严格的要求，因此，中标的本土厂商将受到资金和产业化双方面的扶持。不仅如此，“核高基”项目也对整体 Linux 产业具有重大意义，将帮助 Linux 推广至更多企业客户并有效促进 Linux 产品技术和服务配套整体水平的提高。

2008 年，虚拟化成为 IT 市场热点，客户对虚拟化产品和方案的接受度已大幅提升。Linux 厂商也对虚拟化产品加强了研发和推广力度。尤其是在经济危机的大环境下，Linux 的虚拟化产品与其他主流虚拟化产品相比显示出了明显的低成本优势。目前，Linux 虚拟化的产品已开始在中国的一些电信、政府和教育行业客户中得到了应用，这些成功案例的示范效应将促进更多行业客户接受 Linux 虚拟化的产品。

近几年来，客户对于 Linux 项目的专业服务越来越重视。尤其是高端的电信或银行客户，正在将更多更关键的业务部署于 LinuxSOE 之上，出于保证系统可用性和高性能的需求，他们越来越多地投资于 Linux 厂商提供的专业服务。相应地，Linux 厂商也在修炼内功，提高自身的服务能力。Linux 的专业服务逐渐被重视和认可，这些都促进了 Linux 项目实施的满意度提高，从而推动 Linux SOE 在企业中的应用推广。

总而言之，虽然 2008 年下半年开始迅速蔓延的全球经济危机给 IT 市场带来了极大的冲击，但是在经济下滑的大环境下，Linux 产品的低成本优势将愈发凸显，获得更多的关注和青睐。并且，中国政府已经出台了 4 万亿经济刺激计划以及十大产业振兴计划，会对 Linux 在交通、医疗、制造等传统行业的推广起到很大的推动作用。因此 IDC 认为，在未来几年中国 Linux 市场将迸发出更大的潜力，保持高速增长。

Novell 公司 Linux 业务仍然处于亏损状态

Novell 一直为他们的 Linux 事业部门在做投入。现如今经济形势不容乐观，Novell 虽然在想法设法提高他的 Linux 部门效益，但是他们是否能够盈利呢？依照他们的 CEO 和 CFO 的话说，现在还没有。

在最近的 Novell 公司季度会议上，公司 CFO Dana Russell 认为，虽然公司在 Linux 部门投入了大量的人力物力，用来夺取 Linux 市场以及获取新用户，但是，目前的 Linux 部门还没有盈利，从现在开始，12 个月或者 18 个月后，也许是 Novell 公司 Linux 部门取得突破，开始盈利的转折点。为此，我们已经做了充分准备并且制定了相应的工作计划。

Novell 从 2003 年开始涉足 Linux 操作系统领域。当时他们花费了大约 2.1 亿美元用来收购德国的 Linux 发行版厂商 SUSE Linux 以及美国的 Linux 桌面操作系统供应商 Ximian Linux。Novell 周四发布了 2009 财年第二季度财报。财报显示，Novell 利润为 1560 万美元，每股收益为 5 美分。去年同期 Novell 利润为 590 万美元，每股收益为 2 美分。第二财季 Novell 营收下滑了近 9%至 2.156 亿美元，低于分析师预期的 2.18 亿美元。去年同期 Novell 营收为 2.357 亿美元。Novell 表示，维护和服务订购业务营收增长了 5%至 1.583 亿美元，软件许可业务营收下滑了 32%至 3030 万美元，专业服务、培训服务和技术支持业务营收下滑了 33%至 2700 万美元。

值得一提的是，这个季度中 Novell 公司的 Linux 平台产品表现抢眼，销售额达到创纪录的 3700 万美元，比去年同期增长了 25%，如此大规模的增长可能得益于 Novell 和微软公司的合作，因为微软公司在卖互操作环境解决方案的同时，也帮助 Novell 公司卖出了大量的 SUSE Linux 操作系统。

Novell 公司 CEO Ron Hovsepian 认为，Novell 公司 SUSE Linux 业务的大幅增长也有部分功劳属于渠道的拓展，他很高兴看到，无论从质量和数量上，通过合作伙伴的方式得到的签单用户用户越来越多，而且他希望通过这种销售方式得到的增长能够得到继续。而且 Novell 公司也在不遗余力的将公司以前的 NetWare 客户转移到基于 Linux 的开放企业服务器（Open Enterprise Server）上来。OES 可以为用户提供并行的虚拟化解决方案，而这一切正式基于 SUSE Linux 之上。另外，Hovsepian 透露，通过最近的调查显示，有超过 75% 的受访 NetWare 用户已经将部分生产服务器转移到了 OES Linux 服务器之上。

因此我们不难看出，Linux 不仅仅是 Novell 公司的一条重要的产品线，更是其未来发展的战略重点。Hovsepian 认为，Linux 在我们为用户提供综合跨平台业务、身份识别和企业资源管理等解决方案中扮演着基础平台的角色，它是无法替代的。只有在此平台上，Novell 才能够为用户提供各种传统行业解决方案、应用服务以及云计算环境。

此外，在会议上 Hovsepian 还被问及 Oracle 对 Sun 的收购是否影响到 Novell 的业务，他表示，收购业务对 Novell 来说并不是一件坏事，我们要从中看到机会，Oracle 和 Sun 的两拨用户是截然不同的，在收购完成后，他们很难不会做到统一，这就意味着有大量的客户将改换门庭，这不得不说，对于 Novell 这将是一个难得的机会，因为用户总是会尝试着不同的选择。

IBM 高管称 Vista 利于 Linux 部署

据国外媒体报道，IBM 公司 Linux 和开源副总裁鲍勃·苏托尔（Bob Sutor）就全球 Linux 部署调查情况表示，微软 Windows Vista 促进了 Linux 的部署，企业 Windows 用户正在减少。苏托尔表示，这对微软是一种正面的打击，用户意见找到了微软系统的替代品。“不只是 Linux，苹果 Mac 系统的使用率也在上升。”他表示，微软现在处在艰难时期，必须依仗 Windows 7 的表现。



据英国分析公司 FreeformDynamics 的《桌面 Linux：主流商业应用》调查显示，成本是驱动企业部署 Linux 系统的最大因素，超过 70% 的被调查者都把这作为首要原因。此次调查包括了不同国家的 1,275 位 IT 专业人士，90% 的被调查者都有在企业部署 Linux 系统的直接经验。

另外，调查还发现，尽管 Linux 系统的技术要求较高，但 Linux 系统的迁移还是受到一些非技术

用户的欢迎。FreeformDynamics 的研究主管 DaleVile 表示，一些用户对其桌面计算环境十分关注，可能会十分热衷于 Windows 系统。“Linux 系统的推广可能避开这类用户，侧重于那些只把桌面作为一个工具的用户。”

苏托尔对此也表示，Linux 系统已经被各类用户部署，例如电话中心工作人员。据悉，一些企业选择 Linux 系统还看着 Linux 的安全性和低维护成本的特点。苏托尔认为，熟练掌握 Windows 系统技术的人很多，但 Linux 的维护和支持需要更少的人。“安全性、可量测性及成本是 Linux 系统的最大优势。”

Dell 称大部分用户并不需要最新的 Linux 版本

Dell 最新发布的 Mini 10v netbook 并没有将操作系统从 Ubuntu Linux 8.04 升级到更新的 8.10 或者 9.04，因为 Dell 表示大部分用户并不需要最新的 Linux 版本。

Dell 称这样做既可以控制产品的成本又能够构建稳定的平台，而且大部分用户并不太在意操作系统是 [Ubuntu Linux]8.04 还是 8.10 或者 9.04，用户只是需要正确、稳定运行的系统。

不过 Dell 一直通过升级来维护操作系统，而且只有那些确定很稳定的 8.10 或者 9.04 中的性能才会加入到升级包中。

甲骨文 CEO：未来几年将大量投资 Java

近日，甲骨文首席执行官拉里·埃里森和 Sun 董事长斯科特·麦克尼利在 JavaOne 会议上都选择了不讨论 Sun 未来的硬件业务。不过，埃里森说新的甲骨文将考虑制作采用 Java 软件的上网本和手机。



甲骨文首席执行官和创始人埃里森在甲骨文宣布以 74 亿美元收购 Sun 之后首次公开露面。他在旧金山举行的 JavaOne 会议上表示，收购 Sun 之后的甲骨文将继续大量投资 Sun 的 Java。

埃里森指出，我们所的中间件软件都是 100% 以 Java 为基础的。甲骨文在未来几年里将向 Java 技术投入更多的资金，数额会超过任何一家公司。我们打算投资和加快投资以促进向前发展。

Java 是 Sun 的开源网络编程语言，目前应用于全球的大多数 IT 系统、整个互联网和数百万掌上设备中。埃里森和 Sun 董事长麦克尼利在大会的讲台上都选择了不讨论 Sun 未来的硬件业务。Sun 的硬件业务包括处理器、服务器、存储阵列、网络交换机、瘦客户端和许多其它产品。麦克尼利

开玩笑地说，我们今天的重点只谈 Java。这会使我们避开许多麻烦。

埃里森说，Java 对于甲骨文是非常有吸引力的，因为它允许我们扩展我们的平台并且在此基础上制作应用程序。我们的整个下一代商业应用软件将完全建在 Java 的基础上。我们把我们的下一代商业软件称作 FusionSuite 应用程序。这将是建在一个标准的平台上的 ERP 和 CRM 应用程序。我们能够这样做是因为 Java 是开放的和可扩展的。

埃里森表示，甲骨文的工程师在未来几年里将为 Java 做出大量的贡献。

高通重定义上网本采用 Linux 待机超 8 小时

近日，美国高通公司今天表示，下半年使用高通 Snapdragon 芯片平台的上网本产品将会正式推向市场，高通全球市场营销副总裁丹·诺瓦克（DanNovak）今天对网易科技表示，“高通把使用 Snapdragon 芯片平台的上网本称为 Smartbook，与传统的 Netbook 上网本不同，Smartbook 可以支持全天 24 小时在线的，同时待机时间也更长。”

月初在台北召开 COMPUTEX 电脑展上，高通联合了华硕、纬创、英业达等多家 IT 厂商共同展示了基于 Snapdragon 平台的上网本产品，展示的样机均使用了 Linux 操作系统，其中华硕的 EeePC 上网本使用了谷歌 Android 操作系统，重量只有 900 克，保持了低功耗的同时待机时间也超过了 8 个小时。



丹·诺瓦克表示，“目前的 Smartbook 上网本都使用了 Linux 操作系统，但实际上 Smartbook 是完全支持 Windows 系统的。”据了解，高通的 Snapdragon 平台包括了一个完整的 3G 通信模块，可以支持 WCDMA 和 CDMA2000 两种 3G 高速无线网络。

目前，高通已经与全球 15 家制造厂商共同研制超过 30 款 Snapdragon 平台的手机和上网本产品。今年 2 月，在西班牙巴塞罗那举办的 MWC2009 世界移动通信大会上，日本厂商东芝发布了基于 1G 赫兹主频 Snapdragon 平台的 4.1 英寸手机 TG01，该款手机目前已在日本上市。

与英特尔公司的凌动 (Atom) 处理器一样，高通公司的 Snapdragon 处理器也使用了 45 纳米技术，整体待机功耗低于 10 毫瓦，到今年年底，高通还计划推出主频达到 1.3G 赫兹的新款 Snapdragon 处理器，同时该处理器也具备了常用的 2D/3D 图形计算和多媒体功能。

高通表示，Snapdragon 平台是为 4 寸到 12 寸屏幕的移动通信终端设计的，“过去二十多年，高通在移动通信技术方面一直是业界领先的公司之一，每年约 20% 的营业收入都投入到了技术研发中，这个比例在全球高科技公司中也是非常高的”，丹·诺瓦克表示。

高通公司成立于 1985 年，总部位于美国加州圣地亚哥 (San Diego)。市场调研机构 iSupply 的数据显示，高通是目前全球最大的移动通信基带芯片供应商。2008 年，高通全年营收超过 111 亿美元，同时，高通还拥有 1400 多项 CDMA 技术相关专利。

英特尔收购 Wind River 意在 Linux 市场

北京时间 6 月 5 日上午消息, 据国外媒体报道, 针对英特尔周四收购 Wind River 的举措, 有分析师表示, 此举是为了大力加强对使用其处理器产品的 Linux 的支持力度。

一举两得

英特尔周四表示, 将斥资 8.84 亿美元收购 Wind River。有分析师认为, 该交易既可以帮助英特尔赢得在 Linux 领域的声誉, 同时也有助于该公司推动 Linux 在智能手机和移动互联网设备中的应用。Wind River 是嵌入式 Linux 操作系统开发商, 在智能手机等设备的软件设计工具方面处于领先地位。



The 451 Group 企业软件分析师杰·莱曼(Jay Lyman)称: “此次收购不但可以让英特尔增强在 Linux 领域的竞争力, 同时也可让其限制谷歌 Android 以及开放手机联盟所产生的影响力。” 莱曼指出, 英特尔已在支持 Linux 及整合不同的 Linux 版本方面投入了更多资源。该公司正在开发 Moblin2.0, 这是一款为手机和上网本开发的 Linux 系统, 并已在 5 月份推出了公开测试版本。此外, 英特尔还和 Canonical 在 Ubuntu Netbook Remix 上展开合作, Ubuntu Netbook Remix 是一款上网本普遍采用的 Linux 版本。

英特尔的凌动处理器专门针对手机和上网本设计, 最近该公司还宣布将支持嵌入式设备。今年 3 月份, 英特尔宣布与台积电合作, 向后者开放凌动的芯片设计。目前大多数手机和智能手机采用的 Arm 的低能耗芯片, 英特尔正努力在该领域追赶竞争对手。

互补

有分析师指出, 为了销售更多的芯片, 英特尔需要提供软件工具, 而收购 Wind River 为该公司在嵌入式和手机领域提供了亟需的可信度。Wind River 的编译器等产品可以帮助英特尔优化软件, 从而使其在低能耗的 x86 芯片上运行。

莱曼称: “英特尔对手机和嵌入式软件领域很有兴趣, 而这些领域则广泛使用 Linux 操作系统。该公司明显在积极进入软件、手机和嵌入式设备市场, 此次收购符合英特尔的这些战略。”

Mercury Research 首席分析师迪安·麦克卡伦(Dean McCarron)认为, 英特尔最近在推进用于机顶盒和电视等最新产品的集成芯片的开发, 而收购 Wind River 则可以弥补该公司在这一领域软件方面的不足。Wind River 的 Vtune 等分析及优化软件性能的产品及编译器非常适合于英特尔的软件产品。麦克卡伦指出, 编译器是优化在 x86 处理器指令集上运行的软件的关键。

麦克卡伦称, 除了手机和嵌入式处理器外, 英特尔还可以将 Wind River 的技术用于 Larrabee 显卡等高端多核处理器方面。Wind River 已经和英特尔合作以推出针对多核系统的软件工具。

有分析师表示, 英特尔收购 Wind River 将对该公司芯片用于新产品和市场有所帮助, 该公司的客户希望可以从其芯片中获得更高性能。麦克卡伦认为, 此次收购并不针对任何竞争对手。他指出: “此次收购并非针对任何公司, 更多的是英特尔对未来发展的把握和为成功获取必要的条件。”

Novell 广告直指红帽 Linux 国内 Linux 市场竞争白热化？



最近，Novell 在全球各大技术社区和门户开始推广他的 Linux 订阅服务，目标直指他的竞争对手红帽 Linux，这在以前是很罕见的。以前大家看到的 Linux 广告都是以微软 Windows 或者 Unix 作为比较或者影射对象。而在这次的宣传中，Novell 宣称可以给红帽 Linux 用户提供支持服务，并且费用将大大得到降低，只有红帽订阅费用的一半左右，矛头直接指向了红帽。

可能大家看到，在过去，红帽和 Novell 在国内的策略是大不相同的，以前红帽的中国区的总经理是陈实，技术出身。而 Novell Linux 的东亚总裁张先民博士则是在 IBM 工作多年。在离开 IBM 公司之前，他还是 IBM 大中华区 Linux 业务部门的负责人，对中国区市场了如指掌。另外 Novell 的核心团队都是由以前国内 Linux 企业的资深人士组成，再加上 Novell 以前在国内企业界的知名度，Novell Linux 很快在国内大型企业和政府部门中得到应用，比如中兴、华为等。而红帽则是在中小企业得到了更多的支持，红帽传统的工程师文化也是其在国内发展的重要的资源，大家可以在各个社区的讨论中，红帽 Linux 派系的讨论最多，从红帽的 RHEL 系列到其社区版本 Fedora，另外还有 CentOS Linux 等。而大家却很少看见讨论 SuSE Linux 的。

现如今，红帽也开始改变策略，开始重视国内市场的渠道建设，逐渐在大客户方面投入重要资源，管理层也发生了重大的变化，陈实换成了胡柏林，后者是有着三十年销售和咨询经验的 Oracle 老兵，据说在他的带领下，红帽在大客户市场正在和 Novell SUSE Linux 进行争夺，目前已经和华为取得接洽，能否从 Novell 手中抢到华为将成为这个战役的重要转折点，因为华为是 Novell 自 03 年开始在国内拿下的第一个 Linux 大客户。

Novell 凭着他与微软的合作，在国内市场也大打混源部署的牌，在这则广告中也有得到。而今年三月的，红帽也开始宣布和微软进行虚拟化方面的合作，凭借其得到内核支持的 KVM 虚拟技术，再加上和微软的通力合作，Novell 全力支持的 Xen 虚拟化技术能否继续在 Linux 平台独霸下去，看来将是个问号。

戴尔为中小企业预置开源软件包

根据国外媒体报道，戴尔计划在全球内为中小企业提供开源软件和硬件的预置包。

这是很明显的由专有软件向开源软件的转变，尤其是中小企业期望削减成本。Amit Midha，戴尔亚太区中小企业部总裁称，"客户越有超前意识，就越能接受开源，因为他们会问为什么还要在可以免费获取的软件上面花费。"

戴尔已经在美国市场以零售的方式推出该产品，今年晚些将在亚太区推出。除了零售业务，戴尔也计划同样的预置在制造业、医疗、广告页和在线游戏业等商业方案上。包括美国、中国的很多国家，中小企业的电脑需求正在恢复，中小企业部分要比大企业表现的更好，因为中小企业市场反应对经济变化更灵敏。

戴尔的开源软件预置方案受到了华尔街分析师的认可，对开源界也是个实质的利好。

Google 在 I/O 大会上展示 Android2.0

这次在 Google I/O 上面，则是正式对外展出了部份 Android OS 2.0 Donut 的新功能，包括了内置的 Android Search，也就是让使用者可以搜寻 Android 手机内（通讯簿、行事历、音乐、各个软件的内容）以及网络上的各项数据，另外还有语音辨识的 API，让开发者可以将类似 Google Voice Search 的功能整合到自己的软件。



另外 Google Android 团队还展出了通过手势输入来快速搜寻的功能，好比说在屏幕上画个 "e"，就可以跳到歌曲列表中的 "e" 字头（应该也可以应用在其它的软件）；除此之外，使用者接口的部份，并没有太大的差别。

作为主要功能的 Android Search 是一个很有意思的本地搜索服务，也表明了 Google 除了在 Web Search 上的霸主地位外，在本地数据检索上也有相当深厚的功力，当然这两者也是相辅相成的。

除此之外，Google 和 Salesforce 周三在 I/O 开发者会议上共同宣布，双方的云计算服务平台将实现数据互通。通过使用 Force.com 网站提供的库，开发人员可以从 Google App Engine 中直接调用存储在 Salesforce.com 云计算服务中的数据。Google 正试图利用 App Engine 来吸引企业用户使用 Google 的互联网产品，例如 Google 文档和企业搜索。App Engine 能帮助企业 IT 部门方便的实现对 Google 旗下产品的订制，该产品的竞争对手包括微软的 Azure 和 Rackspace 的 Mosso 等。

同时谷歌向出席其 I/O 2009 大会的所有开发人员每人赠送了一部基于 Android 平台的宏达电 Magic 手机及一个月的免费服务，总价值约一百万美元。

谷歌希望通过这种方式促使更多的用户使用 Android 手机，并有更多的开发人员为 Android 手机操作系统研发应用软件。苹果的成功模式已经证明，高质量的手机应用可以成为推动智能手机销售的重要因素。

包括摩托罗拉在内的众多手机生产商非常看好 Android 手机，认为这一手机平台具有巨大潜力。目前来看，谷歌在智能手机市场面临着苹果 iPhone 和 RIM 黑莓等对手的激烈竞争。不过，对于谷歌来说，Android 并不一定要成为公司长期发展的重要盈利工具。通过推广 Android 手机，更多的用户可以使用手机浏览谷歌网页，这有助于谷歌推动其广告业务。

中小企业如何选择 Linux 备份工具？

很多 Linux 数据备份产品都在不断增加新的功能，且开始在企业中变得更为主流。现在，多数主要存储管理厂商（如 HP 和 Symantec）已经为其存储管理工具提供了 Linux 版本。在一些 Linux 数据备份软件中，厂商们已能够提供备份到云端、处理虚拟化系统和数据去重的能力。显然，现在寻求备份 Linux 系统的用户有了更多的选择，可以享受更先进的功能。

随着云备份变得更加普及，Zmanda 等 Linux 备份厂商正在向他们的产品中添加 Linux 云备份。以 Zmanda 为例，Zmanda Enterprise 备份产品允许从 Linux 设备备份到 Amazon 的 S3（Simple Storage Service）云。

同样，Linux 数据备份产品也提供对虚拟机的支持。随着虚拟化应用在各类服务器（包括 Linux 服务器）上的起飞，这种能力在过去两年里呈现出爆炸式发展。虽然虚拟化的系统可以利用任何备份系统（包括 shell 脚本）来备份，但有很多最适合用于虚拟化备份的软件。例如，Symantec Veritas NetBackup 使管理员可以拍摄一台物理服务器的单一快照和捕获所有运行在这台服务器上的虚拟化服务器，而不必拍摄每一个虚拟服务器的单个快照。

基于 Linux 的虚拟化备份产品通常将任何东西，从虚拟化的应用上的单个文件到整个裸恢复，恢复到物理或虚拟 Linux 服务器。它们还可以克隆虚拟服务器的已有实例，来创建另一个虚拟服务器，为进行快速恢复提供了方便。集群支持在 Linux 备份产品（如来自 BakBone Software 的 NetVault: Backup）中越来越常见。这种支持包括为进行备份和存储管理从控制台上的单一屏幕作为单一设备来管理集群的能力。

NetVault 代表着 Linux 备份软件领域中的另一个趋势。它通过一种叫做 NetVault: Replicator 的特性提供继续的数据保护（CDP）。这种特性通过 LAN 或 WAN 复制字节级数据变化。数据去重也通过 Data Domain 等公司进入了 Linux 备份功能中，这些公司将数据去重技术作为产品特性的一部分来提供。当然，同任何存储管理特性一样，不同的厂商在 Linux 备份上也是有优劣之分的。在选择 Linux 备份产品时，必须进行测试，以确保候选产品可以满足自己的需要，而不只是填写特性清单上复选框的工作。

不过，如果用户只是寻求一些基本的备份功能，市场上也有很多可以提供基本备份功能的 Linux 产品。其中的一些产品只是依靠像 tar 这样的 Linux 实用工具来处理备份任务的简单的 shell 脚本。另一些则是像 Bacula 这样的，可以提供更多额外特性的更复杂的产品。

欧盟欲通过扩大保障软件消费者法律 恐分裂开源界

专家表示，如果欧盟执委会真的通过一条扩大保障软件消费者的法律，那么开放源代码软件开发社区可能面临分裂。欧盟执委会主张，软件公司应该对在欧盟销售产品的安全性与有效性负起责任。

Ovum 公司 IT 研究资深副总裁 David Mitchell 认为，这可能强化开放源代码软件厂商现行的商业模式，但却让独立开发商更难出头。Mitchell 接受访问时表示，这项提案可能促使软件厂商强迫顾客在购买软件时，附带签署软件支持与维护合约，以协助厂商履行产品保证义务。

目前，Red Hat 和 Canonical 等开源软件商已经这么做了，他们都销售支持服务。另一方面，独立开发商的车库开源代码模式(garage open-source model)则将受到考验，因为他们规模不大，无法提供那种程度的产品担保。

新加坡 Keystone Law 的经理 Bryan Tan 先前在一篇博客文章中指出，随着独立开发者必须对软件负责，开源软件可能“崩塌”。他预测：“软件可由两个家伙在车库里写成的时代，已经成为过去。”Tan 并表示，欧盟这项法案可能造成欧盟境外的软件价格上扬，因为业者必须加计保险成本。而且，小型软件厂商“衰亡”（以及市场缺乏竞争），也将推升价格。

欧盟执委会认为，这项提案符合消费者的利益。Mitchell 则认为，此举会造成不确定性，“消费者将发现现有的支持与维护合约变得含糊，或与新的法规互相矛盾”，而企业也面临更长的软件测试周期，造成开发计划耽搁。

讲得更实际些，Mitchell 指出，法律责任将难以明文规定，这是因为硬件与软件之间互相依赖的缘故。某软件发生故障，可能归咎于另一种已安装的软件，或是硬体某个部分出了问题。

不过，Allen&Gledhill 事务所合伙人 Stanley Lai 认为，消费者仍会受益。尽管软件价格可能上涨，但品质更有保障。Lai 认为，此时推断开源件将死，是“言之过早和过度简化”的说法，毕竟开源软件源代码是公开的，靠“众人检视”更能轻易修正错误，使用者和消费者或许会倾向通过社区矫正错误，而比较不可能诉诸法律追究责任。

RHCE 十周年：38000 个红帽认证工程师遍布全球

6月2日，红帽 RHCE 十周年庆典在北京交通大学科学学堂正式举办，庆典内容包括现场 Linux 技术较量及主题演讲两大部分。红帽大中华区总裁胡柏林、红帽大中华区培训和服务经理黄健耀，来自世界 500 强的外资企业经理人及诸多 RHCE 获得者出席了此次活动。在庆典现场，同学们及远道而来的开源爱好者齐聚一堂，与红帽专家、职场精英共同交流 Linux 前沿知识，探讨经济危机形式下 Linux 人才的机遇。



在 RHCE 十周年庆典活动沙龙现场，Linux 爱好者进行了激烈的 Linux 技术较量，随后红帽 Linux 技术精英为现场的同学详解参赛题。参加技术较量的同学均表示，红帽认证工程师（RHCE）实战性较强，能够较好地证明个人 Linux 实际应用水平，并期待早日通过 RHCE 认证，在职场道路上快人一步。

在主题演讲会场，胡柏林发表了主题为“开源发展前景”的演讲。胡柏林表示，在经济危机的形式下，美国政府以及各行业均将大幅度采用开源软件以节省成本。2009 年 3 月 17 日，美国国防部防务信息系统局宣布和开源软件研究会（OSSR）签署联合开发和研究协议

（CRADA）。红帽作为开源厂商中的佼佼者自然获得了不俗市场表现，并在“未来十年影响信息产业方向的公司”评选中位列第八。2009 年 2 月 18 日，DLT 与 US DISA 签署了 18,000,000 美金一揽子 RHEL 订阅和服务采购协议，协议有效期长达五年。2009 年 3 月 20 日，CarahSoft 与 USESI 签署协议，为 ESI 提供 RHEL、JBoss、MySQL 技术服务及一年期订阅服务。

黄健耀分享了主题为“中国 Linux 市场走向与人才需求分析”的演讲。黄健耀介绍，在 IT 培训市场份额方面，红帽一直占据着举足轻重的地位，并呈快速上升趋势。调查表明，2007 年红帽是 IT 培训第一阵营的三巨头之一，2009 年红帽已经成为领导区域的唯一。RHCE 一直以来是市场需求最

为迫切的 IT 认证，原因是在团队技能和厂商支持这两大关键要素方面，红帽被认为尤为出色。

随着 Linux 在经济危机形式下的表现优异发展迅猛，Linux 认证也随着大热。调查数据表明，目前 Linux 认证所占比例高达 38.5%，与微软认证不相上下。在 Linux 认证中，红帽认证占据比例高达 61.2%，远超过 LPIC、TLCE、NovellSUSE 的市场份额。与 2008 年相比，2009 年 Linux 认证市场份额增长显著，RHCT 认证由 71.7%增长到 72.2%，RHCE 认证由 68.4%增长到 70.6%，RHCA 认证由 46.7%增长到 67.5%。在“2008 最热门 10 大 IT 国际认证”排行中，RHCE 认证遥遥领先。中国的 IT 认证市场需求曾远远落后于国际水平，但市场潜力巨大。2007 年中国在亚太区 IT 认证排名位居第四，2009 年 5 月已上升到第三位。红帽表示看好 Linux 认证在中国的市场机遇。

现场还有来自世界 500 强的职业经理人与大家畅谈“知识经济对企业的影响”，以及 RHCE 明星分享从菜鸟到 Linux 高手的进阶培训。该 RHCE 明星表示将全力挑战从 RHCT 到 RHCE 到 RHCA 到 RHCSS 的系列认证。凭借优异表现，其获得 2 个月的红帽内部培训，随之赢得红帽实习机会，最终凭借出色表现成为红帽大家族中的一员。

RHCE 认证启动于 1999 年，是以考动手能力为主的最大的 Linux 认证项目之一。目前，全球已有近 38,000 个红帽认证工程师们帮助他们的公司作成功的 Linux 部署和迁移。未来，红帽系列认证将帮助全球范围内及中国的开源爱好者们在开源领域内获得非凡成功。

IDC：开源企业应用软件正在兴起

IDC 对西欧 515 名机构 IT 决策者进行了一项调查，这些机构的雇员均超过了 50 名，其结果显示使用开源企业应用软件的比例惊人。9%的受访者目前使用的是开放源码的后台办公应用软件，而 7%的受访者使用了开放源码的 CRM 应用软件。

调查结果显示，一般说来，各大机构使用开源软件背后的主要驱动力是省去了先期支付的一笔许可费用，降低了总体拥有成本。而让一些机构不采用开放源码应用软件的主要因素是：对软件支持的水平和开放源码软件的质量表示担心。

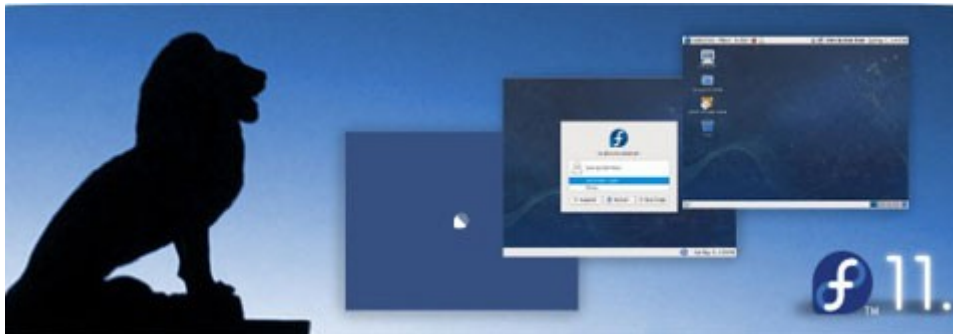
“在企业应用软件市场中，大型供应商拥有 10%的市场份额，开源办公软件采用率为 9%、开源 CRM 软件采用率为 7%已经相当高了，” IDC 公司负责欧洲企业应用程序和服务的研究主任 Bo Lykkegaard 表示，“当然，开放源码的软件可以被某个部门使用，也可以是企业级的部署。我们预计，绝大多数用户在使用开源企业应用程序的同时，也会使用商业的企业应用软件。尽管有所保留，但调查结果表明，在 ERP 和 CRM 领域，开放源代码的使用率已经达到了临界值，应该引起每一个商业软件供应商的重视了，特别是对那些在中型市场进行竞争的供应商而言。”

“我们看到开源企业应用软件的供应商正在吸引股权投资和一些重量级业界领袖的重视，其年增长率通常高于 20%，有时甚至更高。IDC 认为，开源企业应用软件将给对应的商业软件带来价格压力，特别是在中型市场。我们预计，在开源社区和商业软件支持者之间不会出现一场宗教战争 (religious war)。但是，他们可能会在某一方面竞争得很惨烈，比如说谁能够为用户提供 1 欧元的 ERP 或 CRM。”

社区扫描

Fedora11 发布 全面支持 ext4 文件系统

Fedora 项目，一个由红帽公司所赞助开源社区，于 6 月中旬发布了最新版本的 Linux 操作系统平台。Fedora11 版本代号为“Leonidas”，包括升级的虚拟化功能，增强了图形和音效卡的兼容性，并支持更多的文件系统，特别是 ext4。



虽然 Fedora 通常并未广泛的应用于企业，但它常常预示着红帽公司正在开发的企业版 Linux 的功能。“它代表了先进的 Linux 的方向。” 451 Group 分析师 Jay Lyman 表示。“这并不可能真正的与 Enterprise Linux 发行版形成竞争，但它确实也形成了些许威胁。”

改进的虚拟化功能还包括一个新的控制台和升级虚拟机客户制作向导。客户机可以通过支持 SELinux 更安全地运行，Fedora Linux 的内核元件能够实现强制访问控制和基于角色的访问控制，根据 Fedora 项目介绍。Fedora11 的改善还包括内核模式设置，支持更多的来自 ATI、英特尔和 NVIDIA 的视频卡。也许最明显的改善是由于升级内核模式设置功能，加快启动时间，Fedora 项目表示已经下降到 20 秒。

根据 Fedora 项目介绍，新版本通过改善 PackageKit 支持，加载字体和其他内容更快。新的 Fedora11 支持更多指纹阅读器，并提供新的国际语言。新增加的文件系统还支持更高的器件尺寸和文件大小限制。Fedora11 还配备了简易的 GNU (MinGW) 交叉编译工具，以建设 Windows 可执行文件。其他功能介绍可在其相关链接找到。

随着 Fedora11 的发布，Fedora 项目还发布了测试版的一个新的开发者门户网站，旨在提供一种改进的可与社区成员交流代码和共享信息的平台。新的可定制的论坛，可供公众评论。

SourceForge 开始接管 Ohloh 网站

大名鼎鼎的 SourceForge 公司日前宣布，他们将接管 Ohloh 网站。SourceForge 公司可能大家比较熟悉，旗下有著名的 SourceForge.net 开源项目托管网站，以及富有盛名的科技评论网站 Shashdot；Ohloh 网站大家可能不是很熟悉，他们是由一群来自微软的技术经理于 2006 年创办，这是一个开源项目的类似于黄页的网站，收集了超过 25 万个开源项目的详细信息，每个开源项目的详细信息中包括有编程语言、采用的许可证协议、项目代码大小、开发人员数量已经目前该项目的活跃程度等。这个网站创立的目的在于帮助 IT 专业和管理人员选择恰当的开源软件。

通过收购 Ohloh 网站和其数据库，SourceForge 期望能够更加深入的了解开源开发者社区，并通过他们的免费服务为付费用户提供更加精确的广告服务。

Eclipse 用户中使用 Linux 的数量赶超 Windows

根据 Eclipse 公布的调查数据显示，在开发者桌面和产品部署环境上 Linux 在增长，Windows 在下降。Ian Skerrett 最近在其博客中写道“Eclipse 社区调查的 6 大启发”，让笔者很感兴趣，尤其是第一个观点更是令人惊讶，“Linux 在以 Windows 为代价下做的着实不错！”，看看数据先。

	% of Respondents Using	
	Development Desktop OS	Deployment OS
2007 Survey - Windows	74%	47%
2009 Survey - Windows	64%	41%
Absolute Change	-10%	-6%
Relative Change	-14%	-13%
2007 Survey - Linux	20%	37%
2009 Survey - Linux	27%	43%
Absolute Change	7%	6%
Relative Change	35%	16%

很长时间保持的情况都是，开发者在 Windows 上开发程序，忽视了程序布置在哪个服务器端操作系统。这个 Eclipse 数据暗示了一个变化正在发生。使用 Linux 与 Windows 的比率是 27 比 64，令人惊讶吧？当然，很多的开发者在 Mac OS X 上，而不是 Linux；重点是“在”和“为”的问题。虽然如此，但是这个数据估计还是会得到微软的关注。

当然，该数据是基于 Eclipse 用户，而没有计算 Visual Studio 开发者（100%在 Windows 上），但是 Eclipse 和基于它的工具被 1/3 的企业开发者使用。也许微软应该做点什么了，就像微软处理 PHP 所做的一样。还有令 Linux 爱好者兴奋的一点是，Eclipse 用户中，产品部署环境上 43:41，Linux 已经超越微软，成为 No.1。

关于 Linux/Windows 的若干谎言

这篇文章主要从 win 和 unix-like 的系统下常见的几个误解分析，主要为了指出若干在计算机教育界的几个错误理解。这些误解有 win 的软件硬件操作系统性能以及图形设计。胡乱写，但是不胡乱说。

这里收集一些谎言，你常见的，广为流传的，但是的确确的谎言。见诸于诸多的报纸杂志，甚至其中不乏专业的书籍报刊，但是，很明显，他是谎言。谎言的产生有诸多原因，有的可能仅仅是误解，但是也有不少是故意人为。他造成危害及其大，对于中国这个计算机技术落后的国家，特别是计算机教育界，混淆了众多基本概念。考虑到很多人并没有受过专业的计算机训练，甚至根本没有好好学习过《计算机操作系统》这节课，所以就打算从谎言最多的 windows 和他的常见比较者 Unix-like 的系统说开去。

在两者的比较中，我们经常发现两者的优缺点罗列对于专业人员和非专业人员是不同的，非专业（和伪专业）人员经常在适用范围，功能，硬件支持等方面搞错，甚至得出奇怪的结论。

首先来说，在操作系统中，图形界面是在交互方面出现的一次重大的改进，但是并非确定了他就等同于界面友善，操作简单等诸多概念。在交互方面，良好设计不等于使用图形界面，图形界面对于离散稀松控制任务和定制任务，初期完成效果良好，但是持久性不佳，而对于非定制的控制任务，以及逻辑控制任务，是效率非常低下甚至不可完成的。比如，程序设计和算法的概念非常广，

我们日常生活中，简单的控制播放视频次序等，一定程度上算作非常基本的程序过程，其在我们脑中的操作步骤，也是算法的一种。如果是图形界面，事先设计（并且良好）的软件，我们才能够感觉到方便。但是不能等同于，我们只做这么一种任务，我们不需要其他的编程，如果设计到复杂的视频控制逻辑，比如播放条件，播放范围，界面控制同时定制，那么我们就需要记忆大量的空间逻辑（并且，这个还与实际逻辑并非对应关系，所以属于无用的记忆）来完成：某个按钮的点击，某个窗口的拖拉等。我们的系统早就已经批处理了，可以不浪费硬件资源，但是图形界面的控制方式，对于这种任务，又回到了操作系统最原始的状态，只是在后台低度实现了作业批处理。所以，任务的简单性和友善性与否并非和界面有关，而是和任务内容直接相关，一个资深的 linux 玩家（仅仅是玩家我说），可以做到随心所欲控制播放流程的地步，而 win 的玩家只会说他不需耍，其实不是他不需耍，即使需耍，因为 win 的天然图形强关联的弱点，使得他不可能很好的完成。这也是对于强控制领域，比如生产，科研，服务器中 win 始终无法和 unix-like 系统比的原因之一。

所以，unix-like 的操作系统做到的是支持图形界面，但是不放弃字符界面，这使得很多任务的完成成为可能。这仅仅是从用户界面来说的简单例子，只是先要让大家明白，任何的专业课本也不会把图形界面当作什么“先进技术”。

那么下面就说到了硬件支持。如果按照专业观点，世界上计算机并非是指个人 PC 机，在怎么说，cpu 也好，内存也好，磁存储光存储等等，都是属于一大堆无机物诸如石头金属或者沙粒般的东西，没有生命。而我们的计算机艺术就是让他们 clay to breathe 的东西。于是从结绳记事，到计算尺，抑或中国算盘，沙盘，或者西欧的莱布尼茨计算机，后来的巴贝奇差分机等等，都在计算机历史课提及（我国很少有人知道这个课程）。因为从原理上，他们都是一样，都是一个无机的机械计算来完成逻辑的过程。即使是电子计算机，也有很多种，厂家罗列世界各地，各种操作平台从空间站服务器到我们的电子表，不是只有 PC 机一种。但是我们很多把 PC 机技术等同于计算机技术，所以没有人去考虑计算机结构，也没有人考虑 TTL 设计，但是，这些都是很重要的。

win 号称硬件支持广的误解就是 PC 和计算机的概念混淆的结果，其实从现实来看，win 支持计算机软件非常有限，只有若干平台，而且因为受到法律限制，如果没有相应的软件，不可能通过技术手段单独实现（也就是你必须购买获得，当然，中国盗版很厉害）。NetBSD 是 BSD 家族的一种，支持有 120 种以上的平台，普通 linux 支持大约 20 种平台，而 Win 支持不到 10 种，怎么也不可能说 win 的硬件支持比 unix-like 的广。我们所说的广，应该是 pc 的硬件支持，而且还要限制，只是商业硬件支持（想想研究机构和校园，有多少就是自己搭建的）。但是即使如此，我还是知道很多只能在 unix-like 下而 win 下不能使用的硬件，这个我不想多说。很多计算机专业学生中把计算机错误等同于 PC 就是那句盖茨想让每台电脑都运行微软的软件的梦想闹的，盖茨哥哥当初开心是看到了个人计算机的未来，他的梦也仅仅是在个人计算机那里。

因为计算机的概念再次被缩小，所以我们市场杂志上的《计算机爱好者》《程序员》等等很多提到的计算机也仅仅是非常狭隘的概念，至于《电脑迷》等大众读物那些更是，但是我们却认同甚至作为真理，不是可笑么。

硬件的狭隘到软件观的狭隘，我们经常更容易得到 win 比 unix-like 系统的软件多的可笑结论。其实微软的软件来说并不多，因为他的硬件适用范围非常有限，软件会多到什么地方去呢？而且，去 sourceforge 看看那里软件多少（还有多少是没有公布的），在比较一下微软的，两者绝对不是一个数量级。我知道很少人去那里，但是事实看不到，不能于不存在。

也许有人要和我争辩软件和程序的区别了。因为他们不会把 unix-like 下的很多成为软件，当然，我也不想。但是两者我觉着不存在绝对的优势关系。

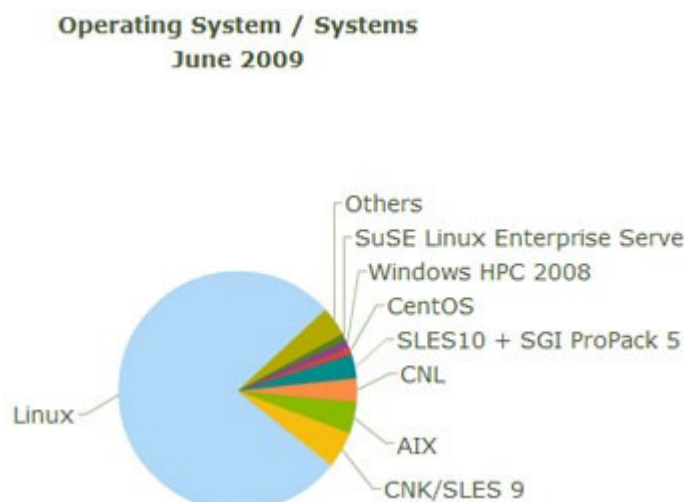
很多人喜欢微软的软件工程学，这个不等同于 unix-like 下的软件艺术或者软件文化(注意，unix-like 也注意软件工程学，但是不光如此而已)，过分的夸大甚至相信了这即使即使即使个学科。unix-like 的程序设计本身就提倡模块化什么的，微软的怎么看都是抄袭来的东西，而且做的并不好。我们看到太多微软的软件大而无当（我承认也有 unix-like 的软件如此），而且如果真的那么好，软件工程要解决的问题并没有使微软任何一个软件 bug 控制达到 unix-like 下软件的水平（当然，不要随便就比，有很多 unix-like 的软件根本得不到反馈，也拿来比较明显不合适），而且，微软软件延迟发布的事情我们听的太多了。

Pc 机和计算机的混淆，软件类别的争论，甚至再说些俗气的，市场占有率的谎言。既然上两者都并非很明晰，所以市场占有率也是一盘浆糊。微软在 pc 市场占有率有优势，即使 MAC 的内核是 BSD 的以后（即使不是 BSD，也不会是 win），还有 linux 的发展，再有其他微量的其他操作系统用户，微软确实仍旧是优势者（但是绝不是绝对优势）。我们有太多的 PC 机干的不是 PC 机应该做的事情，游戏，终端等等，所以 win 有优势也是应该的。但是正如所说，PC 不等于计算机，在计算机界，win 有太多不能进入，有太多劣势，所以在计算机行业，微软并非什么市场的主导。我看到太多的学生立志学习计算机，最后却成为了立志学习 pc 机，立志学习微软，立志做一些琐细的事情，觉着真是可惜，白白糟蹋了太多人才了。

好了，写了这么多，累了，就这样，只希望，别再有太多学生来问：“为何...为何...”了，课本虽然编写的不是百分百正确，但是比杂志还是有看头的多，什么权威的评测都是胡说，如果算法不好，操作系统的内存管理还是会很慢，磁盘碎片还是一大把，而这些怎么是怎么一回事，成因和解决方案都在课本上！浮躁的心加上撒谎的社会环境需要自己的判断，别跟着伪权威搞鱼摸虾，青春本短，浪费可惜。

多数超级计算机使用 Linux 操作系统

过去，开发人员为超级计算机定制各种各样的操作系统，超级计算机在操作系统的使用上没有形成统一规定。近年来，这种情况有所改变。Linux 系统越来越受欢迎，被很多超级计算机采用。但 Linux 在超级计算机中究竟有多受欢迎呢？看了下面的分析你一定会大吃一惊。



Top500.org 定期发布全球速度最快的超级计算机 500 强排名列表。昨天该网站发布了最新一期（两年为一期）的排名表单，我们查看了完整表格并对其中速度最快的前 20 台超级计算机所使用的操作系统进行了调查。调查任务不算轻松，但结果相当耐人寻味。

全球超级计算机 20 强中，有 19 强使用的是 Linux 系列的操作系统

Linux 在全球超级计算机前 20 强中的市场占有率达到 95%！如果只观察前 10 强的话，我们会发现 Linux 的市场占有率是 100%。但如果涉及到整张排名表（全球超级计算机 500 强），Linux 的占有率可能会有所变化，由于时间原因，我们没有一一分析所有 500 强的操作系统。但毫无疑问的是，Linux 已经是大多数超级计算机操作系统的不二选择。（更新信息：在全球超级计算机 500 强中，Linux 的市场占有率为 88.6%。）

可以在 Top500.org 上查看 2009 年 6 月公布的全球超级计算机 500 强完整列表（无操作系统信息），从中我们可以看出，Cray、IBM 以及 SGI 等超级计算机开发商都是 Linux 的忠实拥护者。从表中还可以看出，在 Linux 系列中，SUSE Linux (SLES) 似乎更受超级计算机青睐。

少数情况下，Linux 与运行在计算节点（超级计算机用于实际运算的部分）上的操作系统内核共存于超级计算机中，但这些操作系统内核基本上也建立在 Linux 操作系统的基础上。例如 Cray 系列超级计算机运行的操作系统就是开发人员改良后的 Linux，这种操作系统被称为 CNL（计算节点 Linux）。

超级计算机为什么选择 Linux？

IBM 一直将注意力放在自己的 UNIX 系统上，但近年来也成为 Linux 的铁杆粉丝。IBM 启动 Blue Gene 系列超级计算机开发计划时，就选择了 Linux 作为 Blue Gene 的操作系统。

下面一段话引用自 IBM Research 的 Bill Pulleyblank（摘自 2002 ZDNet），这也许是对各大超级计算机开发商选择 Linux 的最佳诠释。

Linux 是一款开放的操作系统，我们相信经过改良，Linux 能够在 Blue Gene 系列超大型超级计算机上运行。我们可以从 Linux 的开源社区中获取社区成员对 Linux 的贡献与反馈，这是我们可预见到的 Linux 的强大优势。看来 Linux 几乎完全攻克了超级计算机市场。

红帽 Fedora11：颠覆 Linux 难用说

在过去几年中红帽公司备受人们的批评，原因是其重视更有利可图的企业服务器市场而忽略个人计算机市场。这种批评并非没有道理，数年前红帽就已经决定成为一个企业软件公司。

但是，任何认为红帽在赢取企业客户的过程中忘记个人市场的人，需要尝试一下 Fedora11 的最终版，这是红帽针对台式机和笔记本的社区版操作系统。在过去一周中我一直在试用 Fedora11，发现它非常完美和专业，在关键性能方面甚至达到或超过了 Windows 操作系统。

从 Fedora11 的功能列表中，你可以看到它的默认文件系统是 ext4，你或许对此感到不解，但我并不关心操作系统底层的東西，只要好用就可以。

实际上，易用性正是 Fedora11 的最大卖点，当然还有更快的运行速度，从加电到登录系统仅有 20 秒或更短的时间。这一点上完全打败了 Windows。（如果 Windows 希望在启动性能上赶上 Linux，它必须去求助于 Linux，像 DeviceVM 的 Splashop）但是，这是我在几个 Linux 发行版中所感受到的共同点，包括 Moblin Beta 2、Ubuntu 9.04 Netbook Remix 和 OpenSUSE 11.1。尽管没有哪一个堪称完美，不过我喜欢的 Mac OS X 和 Windows Vista 或 XP 也同样如此。

实际上，作为测试的一部分，我一直把基于 Linux 的上网本放在家中，让我的孩子和妻子可以随时按照自己的想法去使用它。我妻子在互联网电影数据库 (IMDB) 上了解自己喜欢的演员。我的女儿在上面写论文。她们都可以轻松的完成这些简单的任务，诸如设置无线网络连接等。一切都非常容易

用，与 Mac 或 Windows 用户的操作习惯非常类似。

这就是“桌面”Linux 目前的现状：它不需要再去证明什么。它花费了数年时间来实现用户友好，现在它已经实现了这个目标，当然这也受益于基于浏览器计算的全球化转变。在这一点上，Fedora 和其它 Linux 版本能做的唯一事情就是接受和扩展 Windows 或 Mac 操作系统的计算体验，因为它们在这方面已经做得非常好(尤其是 Windows)。

如此说来，Ubuntu 的 Mark Shuttleworth 已经将 Mac 作为其要击败的“桌面”操作系统目标，就不奇怪了。

我对 Fedora11 唯一不满的地方是，它至今没有一个上网本专用的版本，而且并不是只有我一个人在期待“Fedora Netbook Remix”这样的版本。

另外，通过使用 MinGW 环境，从 Fedora11 中就可以直接编译以及调试 Windows 程序的全部功能。而且它还内置了最新最优秀的开源软件，诸如用于浏览网页的火狐 3.1 版。

不要再认为它是神秘/令人讨厌的，Linux“桌面”的那个时代已经过去。它依然需要完善，但 Windows 也是如此。苹果的 Mac 操作系统在漂亮和易用性方面算是最接近完美的“桌面”操作系统，不过，从 Fedora11 的表现来看，Linux 落后的并不远。

开源社区的认可来自于贡献

作者：胡才勇

2009 年 6 月 12 日，第十三届中国国际软件博览会系列论坛之“2009 开源软件发展论坛”在京成功召开。论坛聚集了来自各地的开源领袖、IT 知名企业、专家学者和国内有影响的开源技术厂家，北京红旗贰仟软件技术有限公司（以下简称红旗 2000）也应邀参加了此次论坛。红旗 2000 总经理胡才勇先生发表了题为《开源社区的认可来自于贡献》的主题演讲，就红旗 2000 的开源策略及与开源社区的互动等做了详细的论述。



本次论坛的主题为“创新应用促发展--经济低迷时期的开源产业机遇”，重点围绕开源软件、云计算、MID 等最新进展与应用展开研讨。就开源策略问题，胡总在演讲中强调的一点是“来自社区、回报社区”，红旗 2000 按照 LGPL 协议的要求把自己的工作直接回馈给社区，使自己成为一个真正开源的企业，从而通过自己的开源，促进 OpenOffice.org 的发展。

红旗 2000 对社区的贡献业界有目共睹：红旗 2000 是 OpenOffice.org 中文版本的管理者和中

文社区的建设者。目前，由红旗 2000 主导推出的 OpenOffice.org 简体中文版本下载量急剧增长，已成为 OpenOffice.org 产品下载增长最快的版本之一，根据社区提供的数据表明，OpenOffice.org 产品下载超过 1 亿，其中简体中文版下载超过 2000 万，已有 2000 万用户从红旗 2000 开展的简体中文版本工作中受益。由红旗 2000 研究的 UOF 过滤器，已成为 OpenOffice.org 产品解决 UOF 和 ODF 的转换问题的唯一解决方案并被很多企业 and 用户选用。红旗 2000 的开源工作深入到社区的每个技术层面，已成为 OpenOffice.org 社区的第二大技术力量，也是第一个在国际开源社区取得领导力和影响力的中国企业。

作为国际 OpenOffice 社区的主要参与者，红旗 2000 的努力也得到了国际开源社区的认可。红旗 2000 技术副总经理金友兵博士已成为 OpenOffice.org 社区的高级顾问团成员；开源技术部总监成修治已进入“工程师领导委员会”；文档互通技术总监理林已成为中文本地化项目组负责人；其中系统研发部总监李恒成为 OOo 性能改进项目组负责人……红旗 2000 为 OOo 社区提供了一支专业的开源队伍，已有近 50 人参与到社区中去，负责和参与了十几个 OOo 项目。红旗 2000 还将参与到 OOo 产品的设计中，基于中国市场的全面创新理念得到了社区的广泛关注和认可。

在开源活动方面，红旗 2000 以顶级赞助商身份参加了巴塞罗那 OOo2007 国际年会，加大对社区的贡献和参与力度。2008 年 3 月 3 日，红旗 2000 公司以绝对优势赢得 2008 年 OOo 年会的主办权。第六届 OOo 年会于 11 月 5-7 日在中国北京举行，来自全球的开源顶尖人才汇聚北京，共商开源未来发展。这是 OOo 年会首次在欧洲以外的国家举办，充分表明 OOo 社区对红旗 2000 参与开源活动的高度认可。

红旗 2000 目前是国内最大的基础软件企业，也是在国际上贡献最大的中国开源企业，极大地改善中国企业只索取不贡献的形象。特别是通过第六届 OOo 年会的举办，红旗 2000 的全力投入，及中国对开源的支持和各界开源爱好者的热情参与都让社区进一步增强了在中国推广开源技术的信心。社区各主要负责人都纷纷表示此后将加大对中国开源事业的关注程度，中国将成为开源运动的重要区域，也是开源技术发展密不可分的一部分。

Linux 之父：2.6.30 内核将带来“新世界秩序”

Linux 内核 2.6.30 已经放出，与上一版本相比具有数百个变化，其中包括一个新的挂起/恢复(suspend/resume)体系架构，Linux 之父李纳斯托沃兹(Linus Torvalds)称，该功能将让内核迎来新世界秩序。

托沃兹向 Linux 内核邮件列表宣布，通过对挂起/恢复中断请求的重新架构设计，很有希望进入一个新世界秩序，当然我认为众多细节依然需要修改。托沃兹表示，我相信我们缺失了一些东西，而且我们有一些尚待修补的不足之处。不过总体来看它非常不错，最近几天我们已经修复了一些缺点，未来肯定会有 2.6.30.x 版本。

内核的重大版本包括 2.2、2.4、2.6 等版本，更多的功能正在被加入到较小版本中，2.6.30 也不例外。据 Linux 内核版本 wiki 网站 Kernelnewbies.org 称，2.6.30 增加了许多对文件系统的支持，包括基于对象的存储设备文件系统 NILFS2；本地缓存 NFS 数据的缓冲层；在集群各服务器之间分发可靠连接的 RDS 协议；分布式的网络文件系统(POHMEIFS)；和在 ext3、ext4 和 btrfs 重命名文件的自动刷新。

托沃兹表示，和以前一样，在真正公布其详细信息之前我将再等待一两天，我希望人们真正测试这个版本内核。除了文件系统之外，另一个变化较大的方面是设备驱动，具有数百处修改，以及对存储、图形、网络(包括无线)、声音、数字电视、USB、FireWire 和蓝牙设备的增强支持。

如果设备支持基于内核 2.6.30 的 Linux 版本的话，它将为终端用户带来更好的体验。该内核还包括分阶段驱动(staging drivers)支持，目前处于早期开发阶段，尽管这是一个试验中的功能，还是被包含在源代码树中，以希望能借此推动它的开发和稳定。CPU 开发围绕 X86 台式机和服务器和 ARM 移动架构。

革命到底：Linux 系统何时走向完美？

Dennis Walters 是南印第安纳大学 Linux 用户组的一名成员，他喜欢把计算机比作汽车，需要一个方向盘、车门、车闸、车座。最重要的是，需要发动机让它动起来。Linux 就是让 Walters 的计算机动起来的那个引擎。

Linux 是一个开源的计算机操作系统，越来越多的电脑用户开始使用 Linux 了。在企业内部、代理点及大型单位内部，Linux 非常流行。据 Walters 所言，Linux 的个人用户也呈增长态势。

稳定收益

当然了，还有许多别的操作系统，比如微软的 Windows、UNIX 和苹果等。但和这些操作系统不同，Linux 是一个开源的系统，它是一款真正的自由软件，任何人都可以使用它。除此之外，Linux 还有其它的优点。Walters 说：“它比 Windows 要稳定得多。”他所获在过去的两年中，因为出了问题而不得不重新引导或重新启动 Linux，他只遇到两次。而在使用 Windows 的时候，平均一天就要重启两次。

Linux 也非常容易配置。如果你喜欢 Windows 的风格，可以把它配置成看起来更像 Windows；如果你偏爱苹果操作系统，那么你可以把它配置成苹果那样的。Walters 说，你想让它是什么样子，就可以让它是什么样子。“开源”，意味着任何人都可以接触到 Linux 的源代码。这样在你的计算机出来操作系统方面的问题时，你就可以很快地做出响应。Walters 说，病毒和蠕虫也很少会骚扰 Linux 系统。

现在，Linux 已经快要 11 周岁了。它最初是由芬兰的一名学生 Linus Torvalds 创建的，它来自于 UNIX 系统，但比 UNIX 更小，要求的设备也便宜些。Linux 非常容易改变，所以现在出现了成百上千的公司，开发自己的 Linux 版本。

数量在增加

虽然使用 Linux 的计算机用户数量还很小，但这个数字一直在飞速增长。Walters 说：“曾几何时，它还不太成熟；但就在刚刚过去的两、三年中，我认为，它已经成长为一个强健的系统。”

Walters 说，一些大的技术公司，包括 IBM 和惠普，所提供的支持对 Linux 的发展帮助很大。另外，开放源码开发实验室(OSDL, Open Source Development Labs)这个 Torvalds 为之效力的公司一直的坚持不懈地开发并提升这一操作系统及其配套程序。并且，Linux 开源的属性使得用户可以对系统的提升贡献自己的力量，然后和其他用户分享。用户们互相帮助，用户的数量在增加，产品也就得到了提高。

当然，Linux 也有一些缺点。Walters 说：“转到 Linux 上去，其中有一条知识曲线，不过，这种情况和从 Windows 转向 Mac 是一样的。”实际上，Walters 回答了人们挂在嘴边的双关语：你祖母不会因为他祖母使用了 Linux 而使用 Linux。

一个更大的缺点是缺乏商业软件的支持。这就意味着得不到许多流行的游戏和应用。虽然 Linux 中有 Open Office 这样的应用程序可以和微软的 Office 竞争，但还是力度不够。

距离完美还很远

Walter 说，另外一个问题，是缺乏硬件驱动。驱动是用来连接软件和硬件的。不过，慢慢地，获得了一些驱动。他说，如果某个企业想要把 Linux 作为它的操作系统，它通常会雇佣熟悉该系统的员工。如果个人想学习 Linux，可以加入 USI Linux 用户组，Walters 也是这个组的一分子。目前，该用户组大约有 15 位活动成员。

当然了，你还是需要软件、驱动及其它的一些附带产品才可以构建一个好的计算机系统，正如一辆好车需要轮子、仪表板及传输线路一样。但是，Walters 提醒说，Linux 提供了一个非常好的、廉价的引擎，来支撑你的计算机。

行业观察

开源软件的是与非

作者：陆首群

据 IDC 调查分析，截止 2008 年底，全球付费 Linux 服务器（向用户提供支持服务的 Linux 版本）约为 150 万台，免费 Linux 服务器（不向用户提供支持服务的 Linux 版本）则与付费 Linux 服务器基本相当，大约也为 150 万台左右。



所谓“付费 Linux”或“付费开源软件”，指具有商业模式的 Linux 或开源软件版本，而“免费 Linux”或“免费开源软件”，指不具商业模式的 Linux 或开源软件版本。

一般来说，由开源社区开发并发行的开源软件，即我们称之为“开源项目（Open Source Program）”的开源软件，一般都是“免费开源软件”；而由企业（发行商，Enterprise or Distributor）进行二次开发并发行的开源软件，即我们称之为“开源产品（Open Source Program）”的开源软件，一般是“付费开源软件”。

必须指出，开源社区开发和发行的开源软件，首先要体现自由/开源软件的价值观：开放、共享、协同、自由、无偿、选择；所谓开源软件是“利他主义”（或“共产主义”，Altruism or Communism）与“利己主义”（或“资本主义”，Capitalism）矛盾的对立和统一（或平衡），即体现自由开源软件价值观与配置商业服务模式的平衡；企业开发和发行的开源软件，是在继承社区开发和发行的开源软件（价值观、开放源代码、开源许可协议）的基础上，进行再开发（二次开发），并采取相应的商业模式；商业模式是开源软件得以持续发展，实行规模化、产业化的必要条件，开源软件的商业模式是一种以服务为主要内容、实施低价销售战略的模式。

开源软件商业版是基于社区版发展起来的，开源软件商业版与社区版的关系，在实践中可以归纳为如下各种主要类型：

1，双授权模式

如 MySQL。MySQL（社区版）执行自由/开源软件许可协议（GPLv3），其源代码开放，人们可从网上免费下载；MySQL（商业版）是基于 MySQL（社区版）再开发的，软件授权模式发生了变化，MySQL（商业版）是要收费的，或通过提供服务来收费。

又如 Apache。Apache（社区、基金会）是一个非盈利性组织，Apache 软件基金会只向社会提供开源版本（可在网上免费下载），有一些企业如“春之源（Spring Source）”提供支持 Apache 服务或商业版本，但 Apache 基金会规定这些企业不能随意使用 Apache 商标。如果某

互联网网站下载 Apache 软件，安装到服务器（硬件）中，以建立网络/应用服务器系统时，Apache 基金会允许他们称为 Apache 网络/应用服务器，并在其网站上对外提及“Powered by Apache”。

还有如 SugarCRM，也采用双授权模式，其开放的软件（占 80%）不收费，不开放的软件（占 20%）要收费；另外，如 Open Office.org 与 Star Office 的关系，也可归纳为双授权模式。

2，对开源软件社区版进行工程化、产品化处理（二次开发）

如各种版本的 Linux 操作系统（产品）。一般由 Linux 发行商基于 Linux（社区版），进行后续的工程化处理或再开发，完成或发布 Linux（商业版或产品版）；二次开发的工程化技术包含：①优化内核，发行商发布的内核源代码为 Linux 社区发布的“主干（Official）源代码”，允许与其实际执行而不公布的源代码（即发行商的“分支源代码”）有 3-5%的偏差（这是通过测试和经验来获得的），②兼容性测试和质量认证，③对各种软件模块（在内核中约有 1500 个左右软件模块）的不同组合或配置（以提高效率），④回归移植或降级处理（Bug Fix, Patch, back porting），⑤专用工具和插件，⑥本地化需求功能。社区开发的开源软件（开源项目）是原创性技术，企业后续开发并完成的开源软件（开源产品），是开源技术的成熟化过程或产品化过程（二次开发以促使开源软件趋于稳定、优质、高效、成熟）。企业发行的 Linux 操作系统软件（产品），继承并开放由社区开发的源代码，但企业开发的工程化技术是不开放的，含有自主创新因素和自主知识产权，这时 Linux 操作系统作为成型产品的整体软件是要收费的，或通过服务来收费。

Red Hat 的一位开源专家曾说：Red Hat Linux 有两种版本，一种是针对“1 亿人”的免费版本（或收 1 元成本费的基本免费版本），这是为大多数人免费提供的社区版，对这种版本，Red Hat 不提供 Bug Fix、Patch，也不向用户提供支持、服务；另一种是针对“100 个人”（这是指企业用户），每户收费“100 万元”（向企业提供若干套收费的版本，并向用户提供 Bug Fix、Patch，提供支持和服

3，在开源软件（社区版）上采取附加软件模块的方式

如国内开发的“银河麒麟”操作系统，它是在 Free BSD（开源软件）基础上再开发的（自行开发的主要是安全软件，约占整个软件的不足 20%），等于在 80%的开源软件上附加上 20%的闭源软件模块。这样做的结果，连美国国家安全局（NSA）都认为，“银河麒麟的研发使美国向中国军网进行渗透变得无效”。

顺便指出，基于开源软件（社区版），采用附加专用软件模块方式，在二次开发后形成的整体软件中，附加专用模块的构成不得大于 20%，这时整体软件才还可称为“开源软件”。

4，定制模式

如 Ubuntu。Ubuntu 永远是免费的，用户可任意下载、自由使用和修改（这是指其发布的社区版）；Ubuntu 也有商业版，这是指由 Ubuntu 的发行商 Canonical 为用户定制特别版 Ubuntu，Canonical 并向用户提供支持和服务，这是要收费的。举例来说，Ubuntu 支持服务的收费标准为：桌面（每套）9 x 5，250 美元；24 x 7，900 美元；服务器（每套）9 x 5，750 美元；24 x 7，2750 美元。

当然，如 Linux 内核（Kernel）等一些自由软件，是没有商业模式的。

在如上所述的基础上，我要回答一些人对开源软件认识的思想误区（谈一谈开源软件的是与非）：

*有人说：Linux 都是免费的。不对！如上所述：Linux 有免费的和付费的两类；对其他开源软件也如同 Linux。

*有人说：Linux 的缺点是不为用户提供支持和服务。不完全对。只有免费的社区版是如此；而对付费的商业版（或企业版、产品版），Linux 发行商要为用户提供在产品使用寿命周期内全程的支持和服务。

*有人说：开源软件不应是商业软件，不应具商业模式。不对！上面说的很清楚了，开源软件可以配置商业服务模式（只有社区版免费的开源软件没有商业模式）。

*有人问：免费与付费的开源软件在功能与服务上有什么区别？其实，免费的开源软件一般都是原创性的社区版，开源软件（社区版）的性能尚不够成熟，也不能完全满足某些重要用户的特殊需求，它可以免费获得，但得不到开发者的技术支持和相应服务；而收费的开源软件一般都是企业（发行商）后续开发的商业版（或产品版），此时性能趋于成熟（经工程化开发），能更好满足某些用户的需求（做好定制化开发工作），这时用户虽然要付费，但发行商将为用户提供在产品使用寿命周期内全程的技术支持和相应服务。

*有人说：开源软件的开发无国界，是全球志愿者集体开发、合作创新的产物，不能把它限制在某一个企业或某一个国家来进行自主开发或自主创新。其实这说对了一半。开源社区对开源软件进行的原创性开发，是适用于这种说法的，说的完全对！但要使开源软件产品化，趋于成熟，更好地满足用户需求，还要进行工程化、定制化的后续开发工作，这些开发工作一般由企业（或发行商）来承担，这时的二次开发工作当然体现自主开发创新的特点。对于企业（发行商）最终推出的更为完整的、更加成熟的、产品化的开源软件版本（并配置商业服务模式），则可以称之为某企业的版本（并冠以其商标），也可称之为某地、某国的版本，所以，如果说到这种版本的“国产化”，我想也是可以的。

*有人说：开源软件既要体现自由软件的价值观，又要配置商业模式，这是不可调和的矛盾，只能取其一，不能合其二。我要说的是，请你体会一下上面谈到的“开源软件是‘利他主义’与‘利己主义’对立统一（平衡）的产物”这句哲学语言的深刻含义，再看一下上面所列的开源软件社区版与商业版关系的各种类型，这时是否可以解答你的疑惑，而得出这样的结论：不搞极端，做好统一或平衡是完全可能的。

*关于开源和微软的操作系统哪个安全的问题，现摘要 5 月 12 日《华盛顿时报》透露的 9 月 30 日美国国会听证会的有关消息，就一目了然了。《华盛顿时报》发表的署名文章，题目是《中国阻止美国发动网络战争》，文中透露美国政府网络安全专家、国家情报及美国战略司令部顾问凯文·G·科尔曼在美国国会（美中经济与安全评估委员会）听证会上的一些证词，科尔曼说：“美国过去一直利用中国军网和政府网络配备脆弱的操作系统（比如说微软的系统）的服务器的机会进行渗透，自从中国研发了名为银河麒麟（Kylin）这个更加安全的操作系统并于 2007 年服役后，该系统强化了中国关键的服务器，使美国军方和情报机构再也无法实行渗透，致使美国网络防卫能力无效！”

当然，我在这里也要指出，如果只是配置完全开放、透明的开源操作系统软件（社区版），也不足以保证我们网络的安全；网络安全呼唤能将自主创新包含于其中的开源软件。

FOSS 品牌塑造需要克服的七点错误

一般来说，自由和开源软件(FOSS)社区的成员都不喜欢讨论任何关于营销的话题。然而最近很多博客文章开始讨论这样一个话题：FOSS 如何从整体上对外展示自己？这些文章有一个共识，FOSS 还没有建立一个统一的品牌标识，否则她将比现在更成功。

和其他人一样，我也对营销相关的话题不热衷，因为在从事开源事业之前我曾有几年时间进行营销方面的工作，了解其中许多只有局内人才了解的内幕。但是我欢迎进行这个令人不快却日益必需的探讨。

在开源社区内部，FOSS 的品牌已经建立。但是对于开源社区之外的人来说，FOSS 的品牌标识还非常模糊，造成这种局面的原因至少有如下七个，如果开源社区希望突破目前的规模寻求更大发展，必须要首先解决这些问题。

1、任由敌对者建立 FOSS 品牌

目前 FOSS 品牌所面临的麻烦，不是由社区成员自己所树立的。在很大程度上，它是由那些 FOSS 反对者从自身利益出发所建立的。

部分问题是像微软这样的公司多年来宣传所造成的，诸如 FOSS 难用且质量差的说法。这种宣传通常是错误的，或者最多是部分正确的，但是由于缺乏任何有力的公开反击，这种观点已经被多数人所接受。这些说法也把 FOSS 社区置于疲于辩解的状态中，精力被转移，未能制定反击观点来应对完全负面的指责或竞争对手的嫉妒。

然而这种宣传还不是最坏的问题。尽管这些宣传被开源界十分关注，普通计算机用户却可能从未听说过这些东西。相反，更大的问题是，FOSS 对手在很大程度上已经设法完全避免了该问题的探讨。

在很多普通用户心中，Windows 已经成了操作系统的代名词，而像 GNU/Linux 之类的替代操作系统是他们所根本不能考虑的。同样，当苹果推出了其著名的“我是一个苹果机”广告时，FOSS 操作系统却不见踪影。

自由软件基金会欧洲主席乔治·格里夫(Georg Greve)正确的将这些问题看作框定(framing)的一个案例：单方设定争论的条件和语境框架。

就如同一个负面选举广告，设计将对手陷入一场争论，而这场争论中所有条款都是针对对手所设计的，因此 FOSS 的形象让其支持者一直处于防御状态，而不能展示它们自己的任何选择。

2、创建小品牌

正如 KDE 最高领导人 Aaron Seigo 所指出的，在 FOSS 身上没有发生的品牌塑造却通常发生在某个项目或某个企业上。他指出这种现象的原因是项目成员对其成就的自豪，以及让企业客户关注他的公司而非社区的愿望。这些小品牌中有些已经取得了巨大成功，诸如火狐，其余则与成功还相差甚远。

关键所在是，这些小品牌并不突出强调它们与更伟大的 FOSS 整体的联系。举例来说，当 Ubuntu 或 Fedora 等 Linux 发行版在 GNOME 桌面上增加了一个独特的主题或壁纸时，它是在发展自己的品牌，而不是 GNOME。大量的工作被投入到这些小品牌的塑造上，但是它们都太小，无法或很难靠自己树立一个成功的品牌。

或者如 Seigo 所指出的，“由于所有品牌都没有被标识为‘属于同一件事情’，这很难让人们

注意到我们在 Linux 桌面中所做的工作。结果是，我们最终拥有很多小品牌，但是，除了服务器机房或核心 FOSS 社区的人之外，很少有人知道它们。”

作为解决此问题迈出的一小步，Seigo 建议通过 KDE 和其他项目进行合力宣传工作。这种努力能否引起关注现在还不确定，但是关键问题是，如同代码被共享后所有人都会受益一样，建立 FOSS 的更大品牌也将让开源社区的所有人从中受益。相比之下，继续发展小品牌则意味着高投入低回报。

3、分心于细微差别

自从个人计算机诞生以来，用户就支持他们喜欢的软件。但是 FOSS 用户通常是他们所喜爱软件的贡献者，往往与软件具有更大的利害关系。因此，FOSS 用户通常比专有软件用户对软件更加狂热。由此带来的结果是无休止的网络口水战，举例来说，vi 与 emacs，GNOME 与 KDE，或自由软件与开源，类似的争论从来没有停止过。

这些争论的对象的确有实际不同之处，而且这些差异也值得讨论。但是问题是，过于关注这些不同之处，你可能会忘了它们的共同点要远远大于不同点。举例来说，尽管 GNOME 和 KDE 提供了非常不同的用户体验，但它们都是 FOSS，是 Windows 或 MAC OS X 的替代者。

有时候，这个明显的事实会被注意到，于是就有了 freedesktop.org 这样的产品，试图在代码层次上创建不同 FOSS 产品间的协作。但是多数情况下，这种努力在过一段时间后就消失了，争吵再次开始，让品牌塑造的联合努力或任何其它事情成为不可能发生的空想。

一个很能说明此点的例子是，当 Seigo 的提议被链接到 LWN 网站上时，很快被“KDE3.5 和 4.0 之间的相对优点”这个老掉牙的讨论所覆盖，这个话题与联合品牌塑造的努力相比，显然既不相干又微不足道。

4、过于关注功能而忽视其带来的好处

在营销课上你学到的第一件事情是你应该关注产品所带来的好处，而非产品的功能。举例来说，不要告诉客户你的品牌牛奶富含维他命，你只需告诉客户你的产品有益于他们的身体健康；不要详细陈述你的汽车发动机如何进行重新设计，你只需要解释清楚现在汽车如何省油。

一般来说，潜在客户不会对技术功能多么感兴趣，他只想知道你的产品能给他带来什么好处。

自由软件基金会执行总裁 Peter Brown 表示，开源社区往往也犯类似的错误，总是讨论自由软件代码，却不说用户从中获得的价值。他以废物回收为例进行了解释，要想引起公众关注，你不能仅仅谈论熔炉的工作原理和如何融化废旧塑料，你只需指出此举可以帮助避免环境污染。

Brown 建议称，在与普通老百姓交流时，FOSS 需要使用简单战略。谈论代码仅会吸引开发者。在与其它人交流时，FOSS 支持者需要讨论用户对自己计算机的控制权，或产品选择权，以及 FOSS 能够延长其硬件生命的好处。

这些信息都非常易懂，而且更加与普通人息息相关，然而多数时间却被忽略。多数计算机用户本身不是开发者，因此也看不到访问源代码能给自己带来什么好处，因此他们不会被你所说的这些功能打动。

5、没有划清自由和免费之间的界线

到今天为止，FOSS 一直未能区别“自由”与“免费”的区别。说明这种区别时用的最多的话是“自由啤酒而非免费啤酒(free as in speech, not as in beer)”，对于业内人士来讲这是一句浅显而有趣的广告语，但对于其它人却难以理解其含义。

然而，分清两者的区别是非常重要的。不仅仅是因为“免费”远离 FOSS 的精髓，它还可以指那些能够下载使用的任何其它软件，诸如 Adobe 的 Flash 播放器，在这一点上它与 FOSS 同样优秀。另外，它还给 FOSS 带来了“免费等同于劣质”的指责。但是，尽管区分两个含义如此重要，却没有人真正努力的去正确表达它。

6、任由 FOSS 被看作企业战略之一

在自由软件社区内部，FOSS 不仅仅是一个授权。它是一种世界观，是一种运作项目的协作方法，是一种描述用户与硬件和行业之间适当关系的声明。它是社会活动家希望迎来真正社会变革的世界观。

所有这些都是塑造 FOSS 品牌的重要元素，然而却很少被人们提及。至今为止所存在的 FOSS 品牌，仅仅是众多可能企业战略中的一种。

从短期来说，这种方式已经帮助 FOSS 更为接受。但是从长期来看，它可能是一种打击自我的做法。如果 FOSS 品牌完全建立在实用主义的基础上，那些现在采用它的企业迟早会因为发现一个能带来更大回报的战略而放弃它。

我建议 FOSS 应该作为一个整体来考虑更大利益，而不能仅仅盯住作为一个企业战略所带来的短期优势。至少，FOSS 应该把自己打造成变革企业的一种方式，而不仅仅是一个可以被轻易放弃的战略。

7、忘记了最基本的宣传

在品牌塑造过程中所有的困惑和错误行为淡化了这样一个事实：FOSS 实际可能是一个营销家们梦寐以求拥有的品牌。在核心精神上，FOSS 是让计算机更容易使用，确保用户(而非软件或硬件厂商)控制自己的硬件设备。

这是跨越政治界线的一个宣言，更符合我们关于用户支持的观点。它始终如一强调的是简单和强大，这也是 FOSS 吸引成千上万人投入一生来发展它的原因。然而我们现在却很少能够听到它。取代这种富含生命力品牌的是，FOSS 获得了一个模糊而令人困惑的品牌，或者说根本没有自己的品牌。

结束语

我不会假装自己知道如何克服这些错误。它们不是一夜之间形成的，由于反对者的存在和长期习惯的阻力，使得它们很难被克服。我只能说，要想克服它们，或克服我没有提及的任何错误，需要清醒的思考、顽强的意志和大量的交流。

随着 FOSS 的成长，它的社区需要克服对营销的厌恶，采取措施来建立和捍卫自己的形象。尽管营销往往与误导是同义词，我们依然需要借助于它。

而且，当 FOSS 已经向世界提供了如此众多优秀的技术和产品时，在进行品牌塑造时还需要不诚实吗？只要我们实事求是，更好的展示自己并不是误导用户。何况如果 FOSS 社区自己不树立品牌形象，就会让其它人操纵和树立自己的负面形象。

开源非“快速致富之匙”需耐心探寻商业模式

商业周刊曾发表一篇题为《开源：红帽和其它公司的公开问题》的文章，很好的阐述了如何在开源世界中赚钱和迅速成长。

据该文章表示，开源公司在快速成长和盈利方面困难重重，红帽和 Novell 就是很好的例子。

尽管软件供应商红帽的业务成长数据令很多人嫉妒，在截至 2008 年 5 月 31 日的财季，其营收增长 32% 至 1.57 亿美元，利润也有非常可观的 7% 涨幅。但投资者对其并不看好，投资银行业纷纷调低对其股票评级。

问题出在何处？备受批评的是其商业模式：正如新任红帽 CEO 吉姆·怀特哈斯特(Jim Whitehurst)所言，“一个纯服务型企业的防御性不是太好，有些开源公司没有真正看清这一点。”

怀特哈斯特表示，“业界担心我们的增长会放缓，几年来我们一直处于这种困惑当中。”

另外，去年另一家开源厂商 Novell 的日子也不好过，去年其股价下跌 15%；另外一家开源服务公司 SpikeSource 也是惨淡经营。

怀特哈斯特的观点非常准确。开源可以带来金钱收入。诸如谷歌、苹果等公司，通过利用开源，实现了滚滚而来的金钱财富；没有开源，有些东西将根本不存在，诸如互联网、软件即服务(SaaS)和云计算等。因此我们真的需要归纳并找出合适的商业模式。

十多年以来，互联网公司、托管商、运营商、服务数据中心和其它很多公司通过提供基于开源软件的服务，来实现收入。

两年半前，Open-Xchange 决定增加间接的软件即服务商业模式。自此之后它与世界上最大的托管商和许多其它主要托管商成为合作伙伴，今年新增加了 800 万用户。

实现这一点花费了 10 年的时间，因此前 MySQL 公司 CEO 马顿·米科斯(Marten Mickos)可能是正确的：“开源不是一个快速致富的钥匙，你必须要有耐心。”当 MySQL 被卖出的时候，该公司已经度过了 13 年的时间。

但是如果你运用得当，你可能将会像马顿现在一样放松。

盗版软件：不值得冒险

作者：胡铂林

根据反盗版团体－商业软件联盟－的年度报告中指出，中小型企业是持续使用非法软件的用户。尽管商业软件联盟(BSA)作出了不懈的努力，以起诉非法软件的传播者和使用者的方式来杜绝问题，然而 2008 年，盗版导致大约 1.63 亿美元损失。



企业使用盗版软件的一个主要理由是：成本。对于许多企业用户而言，使用许可证方式的闭源软件仍然是非常昂贵的。除软件的购买成本外，在总成本内还要增加技术支持费用和升级费用。对于一个中小型企业，使用桌面工作软件和基本商业应用软件，比如库存管理系统和客户关系管理

(CRM) 应用软件等，这些软件的购买费用，后续的许可证和支持费用可算是一个沉重的负担。

使用盗版软件的商业用户有可能没有意识到他们所承担的风险。他们冒着被法律起诉的风险，当然还要忍受盗版软件的使用质量。尽管如此，闭源软件的高昂费用足以令他们回到使用非法软件。

这正是开源软件具有吸引力的地方。开源软件是一个合法的，经济的选择，来替代昂贵的用户许可证的闭源软件。与闭源软件相反，开源软件是鼓励用户使用，研究，复制，修改和重新发行的软件。该软件可合法地使用于任何数量的计算机上而不受任何限制。就应用上而言，这表明了用户可免除闭源软件的限制和其许可证的费用。

对企业而言，建议他们可购买商业的开源软件，包括了全面的客户支持服务来协助安装新程序或新的安全威胁漏洞。今天，有几百个商业 Linux 发行套件可供选择，能允许用户选择最佳版本来满足其特定的需求。

另外，还有提供给企业用户的全面覆盖服务和产品更新的订阅模式，包括从关键技术的更新，最新硬件和存储环境的驱动程序，到软件功能增强和新架构支持。订阅模式的价值在于每一年提供企业一个可预计的成本结构。用户也不会被绑定在封闭的闭源软件程序的限制性技术架构中。闭源软件要求用户支付许可费用才能获取最新的技术。

开源 Linux 和其订阅模式提供了一个能超越封闭的闭源软件的成本优势。而正因为其成本优势，成为使用盗版软件的企业的一个值得考虑的选择。如果这些用户转用负担得起的开源软件，那么在任何国家零盗版的目标将不会是空想。

现在开源软件几乎能替代所有的主要软件，范围从办公套件和数据库到商业应用。OpenOffice – 开源办公软件套件能为文字处理，电子表格，演示，绘图，数据库提供的功能和性能，可比美任何同等的闭源软件。其它例子包括：

- 开源 CRM 应用程序 SugarCRM
- 开源 ERP 应用程序，如 Compiere ERP 和 OpenBravo
- 开源会计系列 Fitrix
- 开源数据库 MySQL 和 PostgreSQL
- 和 apache 网络服务器

所有这些开源应用程序都等同于相应的闭源软件。通常情况下，用户选择闭源软件的原因是由于对产品的熟悉而非其商业价值。当成本是一个问题时，他们最终使用盗版版本。事实上，许多开源软件应用建立在 Linux 的操作系统上，它拥有的功能和性能相等或优于闭源应用软件。

在性能考虑方面，例如 Linux 有着更优化的源码，在同一硬件规格上，普遍快过闭源软件。它也能够运行在各种各样的硬件平台上，所以我们能较易地完善 Linux 的应用程序运用于工作站，台式计算机，笔记本计算机，互联网或数据库服务器，甚至是在路由器上。Linux 桌面的外观和功能包括图标和菜单，能根据用户要求或偏爱来配置，从而消除任何用户学习一个新界面的问题。

凡使用 Linux 的用户都知道，它是一个高度稳定和强大的操作系统，能抵抗系统毁坏无需重新启动。最好的证明是，开源 Linux 软件领导了主要的商业环境中关键任务应用的部署，如在纽约证券交易所和许多国家政府部门及国防机构。

当企业评估其实际需求时，最重要的诉求是营运时的应用软件运行顺利和支持平稳。开源软件的卓越性能足以应付任务关键型应用，而其经济的成本能令中小企放弃盗版软件不值得冒的风险。

CIO，是时候在企业中部署开源软件了

据统计调查指出：假如不以太严苛的标准来看，开源解决方案在企业 IT 中正日益普及。该调查也同时指出，企业采纳开源软件时，会在哪些方面产生阻碍，以及采用哪一种应用类型才是最受欢迎的。

开源软件解决方案过去都是被公司的工程师在暗地采用；在“得到宽恕比获得允许容易得多”的哲学下，他们偷偷地用着 Apache Web Server 或是开源软件的发展系列工具——“使用开源软件工具，比争取到 IT 预算更简单”，那就更不用提了。

根据 2008 年 4 月时的一份针对 IT 及营运的主管与经理人调查，发现状况和过去已截然不同。从 328 位回复者收集来的数据显示，现在有超过一半的回复者(53%)在它们企业中使用开源软件应用，另外有 10%的回复者计划明年开始采用。而将近一半(44%)的回复者表示，他们也将开源软件应用列入采购流程中考虑，就像专利解决方案一样。

目前被利用的开源软件解决方案，主要运用于操作系统——像是 Linux(78%)、基础架构应用——像是后端数据库与网站服务器(74%)，以及软件开发工具——例如 Eclipse(61%)等等。

此外，在这调查中，有将近一半的回复者(45%)使用了开源软件的桌面型计算机应用程序，例如 OpenOffice.org 软件，而采用开源软件企业应用程序者则有 29%。而那些最受青睐的企业应用程序，则是协作工具、客户关系管理(CRM)工具，以及企业资源计划(ERP)应用等项目。



现在人们对于开源软件解决方案也逐渐具有信心。在 5 位受访者当中，有趋近于 3 位响应者(58%)表示：强烈同意或同意 Linux 对于在担负关键应用方面已经具备可靠度。显然，IT 主管与经理的信心指数高涨：有 62%的响应者表示，Linux 已经准备好迎战最重要的工作时段。

此次调查的回复者包含了 IT 主管或经理(59%)、营运主管或经理(13%)，以及 IT 专业人员(20%)与专业商务人员(8%)。

另一个调查提供读者参考：据美国市场分析机构 Evans Data，针对有关开源软件与 Linux 发展之调查，有 3/4 的软件开发人员(77%)表示，对于担负关键性任务的应用程序采用 Linux，他们有绝对或大概足够的信心。不过，请先对此观点持保留态度：因为取得参与这个市场研究资格的人，几乎都是支持使用或撰写开源软件的人，所以有如此高的比例，其实并不足为奇。

开源软件充满魅力

企业 IT 部门采纳开源软件的主要原因是经济考虑。更低价的整体成本(59%)与采购成本(56%)驱动着这个市场。但是经费不是人们之所以采纳的全部因素；有 32%的回复者表示，更具弹性化才是它被采用的主因，而 1/3(30%)的人认为，可存取原始码是决定采纳的动机。不过开源代码本身的特

性并不是主要的驱动力；只有 12%的人表示，更佳质量的程序代码是被采纳的主因，而有 22%则认为是由产品功能决定之。

尽管这是好消息(至少对它的支持者而言)，有几近 2/3 的企业目前正使用开源软件，或计划不久后会采用，但是在采纳过程仍有许多阻碍。主要的原因是产品支持度问题(45%)；显然，企业希望获得技术支持请求时的响应保证。第二个阻碍则是有关市面上可获得的解决方案，其后续学习的支持仍然不足(29%)；人们希望开源软件应用的学习与利用必须是很容易，可以满足特定 IT 人员的需要；至于其它的原因，安全考虑占 26%，以及管理上的支持性欠佳则占 22%。

此外你会注意到，开源软件应用本身的质量并不是很大的问题。20%的回复者提出，软件质量问题会是主要的阻碍因素，而客制化问题则占 15%。所以，假如你想要说服老板看到更多开源软件优势的话，那么，与其在客制化的问题上努力，不如花更长的时间再度强调，你所使用的开源软件，在技术支持的可得性与质量方面是得到保证的。

整体而言，使用(或计划使用)开源软件的公司都有同样的担忧，就是它们与专利解决方案牢不可分。大部分未采纳者的考虑因素，也正是开源软件的首要症结问题。回复者的数据显示，其公司目前在使用开源软件的占超过一半(52%)。他们指出，产品支持是踏出采用这一步的最大阻碍；那些没有使用开源软件的 1/3(33%)回复者更确切表示，这就是最主要的问题。产品支持仍然是他们最首要关心的项目，只是比较不紧急而已。换句话说，用过这东西的人就会知道问题所在；而没使用过的人只是期待它能够如何。

当企业在考虑开源软件解决方案时，有个项目可以安抚他们：一旦你的内部有了这软件，程序代码的质量问题就会变得不再那么重要。未使用开源软件的企业亦指出，程序代码质量是前 3 大问题(尾随于产品支持与安全考虑之后)；然而，对于那些与该应用长期一起工作的人来讲，它却是第七大阻碍。一旦你开始使用开源软件，你会发现状况比你想象的还要好。

除了上述的问题外，企业要采用开源软件，还有几个问题要排除。根据 Evans Data 公司的调查，企业采纳开源软件的几个最大障碍包括，企业对专利软件的偏好、缺乏接口设备的驱动程序，以及要学习一套新技能的需求等项目。

让开源软件在内部运作

大约有 1/4 的公司(27%)都有适当正式政策，规范开源软件的应用方式。而有 18%预计未来的 12 个月内采纳这类政策。那些有制定开源软件政策的响应者当中，45%觉得他们的政策是很有效的，46%则表示稍微有效。据推测，“稍微有效”的回复者是在思考，IT 人员与开发人员在公司未同意的情况下，不知道已经安装了多少次的开源软件软件：大约每 5 位就有一位(21%)承认这项事实(选择“经常”或“有时候”的人)。

既然现在有超过一半的企业在使用开源软件，那么私下偷用的哲学也就有了很大的变化。企业可能会“经常”(43%)或“有时候”(24%)地视这类的应用为免费软件；他们让这个应用运作，却不去仔细看原始程序代码。虽然可以存取原始程序代码，但是无论他们是否有进行过变更，企业 IT 部门会用他们自己编写的开源软件模块情形，其实并不常见。

有 18%的人经常使用未变更程序代码模块，好像那模块是免费的图书馆一样，而 36%的人会偶尔为之。

尽管如此，将近一半的人(49%)会经常或有时候对错误提出报告，或将他们的变更贡献回去给开源社群；而 11%的回复者表示，他们的员工当中就有开源软件的提交者。

当初如果企业拒绝接受开源软件为适当的应用，显然今天的局势就会完全不同。

开源软件解决方案

有许多企业能够完全地接受开源软件解决方案，但是不知道如何衡量它是否适合其组织使用。提供开放源码商业智能平台的软件商 JasperSoft，其 CEO Brian Gentile 表示，开源软件应该像专利软件一样有类似的方法被评估，例如功能、潜能等。

以下四个准则提供读者作为判断，哪一个开源软件产品或是供货商才适合企业：

- 1、确认该产品已经能从商业开源软件公司取得。有授权机制与足够资源能确保你成功，例如客户支持。
- 2、检查供货商的社群规模与活跃度。这一点会帮助你透过其产品去判断成功的可能性，更胜于从它的年收益角度观察。
- 3、知道该供货商有多少解决方案正为其它营利企业运用。这是一项很重要的指标，你能确保该产品与该供货商真正履行的能力。
- 4、实施标准的评估技术。那也是用在专利厂商上的标准，例如产品质量的发展计划、该企业的人员优势、教育训练的可获得性，以及专业顾问服务等。

开源的成长

超过 60% 的 IT 及商务专业人士开始使用或计划使用开源软件。

IT 领导人受鼓舞，将企业程序代码回馈给开源项目，将组织内部的定制化服务向开源软件社群发表，可为企业带来好处。

即便是在最大型企业内，开源软件也不再是新奇的玩意儿。根据最近的调查，现今有 53% 的企业在使用开源软件。然而，那些企业当中，回馈程序代码给开源软件社群的比例并不多，Red Hat 的总裁暨 CEO Jim Whitehurst 在 Red Hat 高峰会中表示。他同时解释，这样的贡献给企业带来的好处要比为社群带来的发展，事实上是有过之而无不及。

据国际 Linux 基金会执行董事 Jim Zemlin 的看法，75% 的软件撰写是给组织内部使用的。如同 Whitehurst 所指出，有很多程序代码都没被利用的确很浪费资源。“想想在专利城墙的背后，有多少软件被写好搁在那边”。

组织经常会利用像是 Linux 操作系统那样的开源软件，设计出令其它开发人员与使用者赏识的创新技术解决方案。Whitehurst 将它视为，那是在替开源软件能量“传福音”的机会，协助企业更有效运作。

Zemlin 同意，让内部更老练的开发人员涉入的重要性，他将之视为一个尚未被开发的智囊团。“使那些人投入到开源软件的进程是至关重要的，”他说。

举 Whitehurst 所说的例子，Linux 的 Merge 实时信息功能，最早是 J.P.Morgan(摩根大通)为了组织内部需求撰写出来的。但只要公司作任何升级就要翻修操作系统；每次 Linux 计算机升级时，系统都会要求组织重新部署定制化服务。据 Whitehurst 表示，J.P.Morgan 的 CIO 很清楚，透过贡献原始码给 Linux 社群可降低那笔资助成本。其它 Linux 使用者将受惠固然是好事，但是对 J.P.Morgan 来说，更重要的是公司不需要在维护内部应用程序上投资它自己的资源。现在，除了他们自己的员工本身就是提交者之外，Merge 程序代码可以完全地由 Linux 开发人员去做更新与升级。

Whitehurst 表示，同样地，一个加拿大的保险公司开发了 ESB(企业服务总线)基础架构，并且将它贡献给 Linux 社群。他说，透过这种作法建立起一个大型的使用者社群。

这不是极少数的例子。在调查报告中显示，有一半使用开源软件的人(49%)会经常或有时候提出错误报告，或将他们所做的变更回馈给开源社群；11%的回复者表示，他们自己的职员当中就有提交者。

你也许会认为，企业会关注于竞争差异化方面，于是企业会维护各自定制化服务不要落入同行竞争者的手中。实际上，Zemlin 看到的却是相反的。他说，大家都很有意他们的程序代码在 Linux，与其它开源软件中是如何被利用的，而且他们会注意哪个公司作最大的贡献。他们将之视为特有的程序与以表彰，并且认同该企业的专业领域技术。当调查结果出来时，Zemlin 表示，他被问了这句话然后被挂掉电话——“为什么不是我们公司贡献的最多？”。

Zemlin 表示，企业贡献开源软件计划的最有效方式是，当他们自己的利益牵涉其中时。当企业必须为内部的老旧应用或装置驱动器全部重写时，经常都会被升级上的困难度阻碍。Zemlin 表示，“相反地，如果提交变更到 Linux 主线(mainline)状况会比较好，”企业欲开发一个维持水平的系统，无须再为该系统的发展资源孤军奋战。

MIPS 会改变 Android 命运吗？

随着 Android 在移动电话市场的成功，许多公司都看到了 Android 扩展到移动电话以外市场的潜力，这些新一代连接设备需要高性能来满足对多媒体和软件日益提升的要求。虽然业界一直叫喊者 Android 要超越手机领域，但真正推波助澜的事件还是 MIPS 宣布支持 Android 这个举措。

随着 Android 在移动电话市场的成功，许多公司都看到了 Android 扩展到移动电话以外市场的潜力，这些新一代连接设备需要高性能来满足对多媒体和软件日益提升的要求。虽然业界一直叫喊者 Android 要超越手机领域，但真正推波助澜的事件还是 MIPS 宣布支持 Android 这个举措。

上周五，MIPS 于北京召开新闻发布会，宣布推出其行业标准 MIPS 架构的 Android 平台，并希望能在 60 天内公开 MIPS 优化后的源代码。

MIPS 支持 Android，兴奋了谁？

利用 Android 平台，设计者能快速、轻松地移动电话以外的设备带来创新应用和用户界面，这对消费者来说极具吸引力。众多 MIPS 的合作伙伴和授权厂商对 MIPS 支持 Android 感到兴奋，并希望借助此举改变目前的市场竞争格局。

Google 最初借助 ARM 处理器将自己的开源 Android 打入手机市场，目前来看，Android 瓜分手机操作系统市场几成定局。联想近来的热门话题 Netbook，Google 的 Android 系统向手机之外的高端应用转移，无论是厂商还是用户，都希望此举能打破 Wintel 在笔记本特别是移动笔记本领域的垄断。而 MIPS 对此举进行的实际支持，使其有望成为移动高端产品的主流平台。

MIPS 现有的 MIPS3274K 内核和 1004KCPS 的高性能产品也非常适合这个市场。“事实上，已经有厂商把 MIPS 的内核用在上网本中。”MIPS 上海总经理何英伟表示，“如 RMI 的芯片、ICT（中科院计算技术研究所）的芯片，他们都是专门用 MIPS 内核的，而且都用在上网本上。所以从这个角度来讲，MIPS 已经被应用到上网本当中了。”

“对于另一种新兴的 Smartbook，其实有点类似 Smartphone，只是屏幕比较大一些，互动多一些，MIPS 的处理器与 Android 是非常适合这个市场。”MIP 公司亚太区副总裁 MarkPittman 透露，“实际上我们有许多授权的客户和厂商，都特别热衷于进入 Smartbook 这个领域。比如在韩国的 RMI，他们在手持的终端领域，包括 GPS、PDA 等等领域是非常受欢迎的，而且他们在这方面也非常成功。”

传统上，我们都认为 ARM 在手机等便携式领域应该用更广泛，而 MIPS 则在数字电视、住宅网关、调制解调器、机顶盒等领域收获颇丰。针对这种传统观念，Pittman 指出，“便携市场也是 MIPS 公司的目标市场之一。我们有很多针对便携设备的产品和技术，也跟众多专门独立做便携设备生产开发的厂商也有很好的合作。在很多应用方面，如 PSP、移动电视、游戏机、GPS 导航，有很多的厂商跟我们合作，比如珠海炬力、AsusTek 等等。”对数字家庭应用，Kitagawa 更是乐观估计在年底钱将有 18~25 种设备会基于嵌入式 Android 平台。

好处：不仅仅是成本

“目前市场上便携设备的 OS 有很多选择。相对来说，WindowsMobile 价格比较高，而且另外一个问题是它提供的是类似 PC 的一种体验，而 Android 完全不同。”MIPS 公司策略营销总监 KevinKitagawa 表示，“Android 非常的灵活，你可以选择你想要的用户界面，而不需要参考 Windows 或者 iPhone。这就给我们的客户提供了很大的自由度，能够针对不同的市场定制产品。当然其成本优势也很明显。”

有了 Android，授权厂商能更容易的进行互联网接入的产品设计，如图所示，如果没有 Android，很多事情都需要客户自己去做，需要花很多的时间去整个所有的数据，Kitagawa 表示。有了 Android 以后，因为 MIPS 已经优化了很多部分，包括在内核里面，客户就不需要单独开发那些部分，从而节省了很多时间。

如上图所示，“源代码是 Google 提供的，MIPS 对其中一些部分进行了优化，以帮助我们的授权厂商能够更好地利用 Android。类似于你从微软能够得到很多基本的应用，如计算器、网络浏览器等，但是也有很多第三方的软件。这里的情况是一样的。例如，如果你有一部 iPhone，你可以到 Apple 商店里去下载很多免费的应用，还可以去购买很多不免费的应用。Android 也一样，有很多个人和公司都在针对 Android 开发应用。”Kitagawa 表示。

“现在，MIPS 的授权 IC 厂商对于 MIPS 支持 Android 的宣布非常高兴，因为当他们要进入某个消费市场时，用户是不会在意这个设备里面用的是什么处理器。而且使用 Android，他们可以下载很多免费的应用，可以使产品具有更强的互动性。”他补充道。

第一块试金石

网络对消费设备的冲击是普遍性的，我不仅希望手机可以上网冲浪，还希望我的数码相框能去网络上看图片，我的机顶盒能帮我找到网络资源，电视能都将连接到互联网及其丰富的内容，以便给我带来更丰富、便利的体验。

在这次新闻发布会上，Kitagawa 演示了一款家庭媒体播放器(HMP)，该 HMP 设备是 RMI 公司基于 MIPS32 核心的 Alchemy 处理器的一款参考设计、采用了 AndroidOS 的参考设计，处理器频率在 400-600MHz 左右。不仅能浏览相片，视频播放，连 Email 收发、网页浏览、甚至 MSNmessage 的软件都具备。之前获悉 Android 所支持的分辨率 320x480 也得到了改进，在这里支持 1024x768。

这款具有无线联网功能的数码相框，定义在电视、电脑之外的家庭“第三屏幕”，目前还没有任何官方消息得以证实由 OEM 厂商正式拿来生产，甚至在演示的现场，由于没有启动该设备的触摸屏，演示的时候还外接了键盘，因此更是引起了质疑此设备与上网本区别的声音。很明显的是，该设备想要突破消费者的使用习惯，得到其青睐，无疑还得在启动速度、接口和价格方面做出进一步的努力。该家庭媒体播放器的成功与否，目前看来也是 MIPS 公司在手机以外领域推动 Android 应用的第一块试金石。

专家专栏

专注于服务器操作系统的 FreeBSD 是如何炼成的

----国内资深 FreeBSD committer Delphij 专访

ChinaUnix: FreeBSD 是一份 UNIX 操作系统，支持众多的硬件平台，比如 i386、IA-64、PowerPC 及 UltraSPARC 平台。它是基于加州伯克利大学的 4.4 BSD-Lite 发布，并带有一些 4.4 BSD-Lite2 增强。FreeBSD 被遍布全世界的公司、Internet 服务提供商、研究人员、计算机专家、学生，以及家庭用户用于他们的工作、教学和娱乐之中。在国内，也有大量的互联网公司在使用 FreeBSD 操作系统，比如网易和新浪。同时，国内也有少数几位为 FreeBSD 默默作出贡献的开发人员，今天我们有幸请到他们当中的一位----李鑫，他以网名 delphij 活跃于 FreeBSD 项目相关 maillist 和国内各大 BSD 相关社区，比如 ChinaUnix、FreeBSDChina 社区和 newsmth 的 FreeBSD 板块。作为国内的资深 FreeBSD committer，让他给我们谈谈目前的 FreeBSD 项目的发展现状，以及如何参与到类似于 FreeBSD 项目的开源社区中来。



ChinaUnix: delphij，能给 ChinaUnix 的网友简单的介绍一下自己么？

delphij: 我是北京人，现年 27 岁，1999 年经一位师兄介绍认识了 FreeBSD，并从 Linux fans 变成了 FreeBSD fans，大学期间做过一些 Windows 应用和少量驱动程序的开发，大学毕业以后一直在做开源平台上的开发和系统管理工作。我目前在美国硅谷工作，已婚。

我在 2004 年成为 FreeBSD 的 src committer，并在稍后获得了 doc/ 和 ports/ 的相关权限，目前是 FreeBSD 安全小组的成员。我比较关注的领域是基本系统中的工具和 libc 的易用性与标准化，以及内核的兼容性和可靠性，在业余时间不允许我去做比较大块的项目时，我也会花一点时间来改善一部分现有代码的品质。

最早产生参加 FreeBSD project 的冲动是在上大一的时候，很大程度上是因为和 Kirk McKusick 讨论关于 UFS 文件系统的一些问题的时候他提到说 Please encourage continued use of BSD software in your country。当时也感觉到，似乎我们中国大陆的开发人员在 FreeBSD 社区中没有什么声音，而很多东西对我来说并不是很困难，所以就想做一些自己力所能及的事情。

ChinaUnix: 作为国内为数不多的 FreeBSD committer，能给我们谈谈 FreeBSD 项目团队的运作么？以及如何才能成长为一名 FreeBSD committer，最终成为 Core Team 的成员？

delphij: FreeBSD 项目团队是一个非常关注系统设计和代码品质的团队，在技术方面，我们鼓励同僚复审（peer-review）的方法来确保代码品质，并在开发过程中使用包括持续集成（Continuous Integration）和回归测试等自动化的方法来提高由变动所引入的新缺陷的门槛；在管理方面，我们每两年会由过去一年内活跃的 committer 选举产生项目的领导团队——Core Team，而 Core Team 会将自己的一部分技术方面的权力下放给一些专业小组，如负责发行版及其质量保障的 Release Engineering Team、负责管理 Ports 基础设施及管理 Ports Committer 的 Ports Management Team、文档工程小组以及安全长官团队等，而 Core Team 则避免直接参与专业方面的争议处理，作为原则指导及最终仲裁机构存在，其地位类似于普通商业机构中的董事会。

由于美国法律的要求，在项目之外还成立了 FreeBSD 基金会。除了接受捐款之外，基金会还代表 FreeBSD 项目团队进行一些必须以法人名义进行的活动，例如签署授权、持有“FreeBSD”注册商标等。

作为开发团队，我们总是需要补充更多的新鲜血液来适应新的挑战。成为 FreeBSD committer 其实并不困难，找到一个或一些 FreeBSD 没有或不够完善的部分并不断提交相关的代码或补丁，很快就会有 committer 邀请你加入了，当然，FreeBSD 的三个主要的组成部分——内核及基本系统（src）、第三方软件（ports）和文档（doc）都有一定的编码规范要求，这些联机手册 style(9)、Ports 开发手册等都有介绍，成为 committer 的开发人员需要遵守这些规范，并且能够和团队一起工作。

我本人并不是 FreeBSD Core Team 成员，事实上，Core Team 需要关注很多管理方面的问题，因此也就需要在这方面花费一定的精力。成为 Core Team 成员需要在选举期间主动参选提出自己的主张，并获得其他 committer 的支持。

ChinaUnix: ChinaUnix 论坛中很大部分的会员是 Unix/Linux 系统管理人员，能给我们谈谈作为一个系统管理人员，特别是开源系统方面的管理员，成长的过程中需要注意什么？包括技术方面的、学习技巧等等。

delphij: 我个人认为作为系统管理员是需要了解很多基础知识的，这个其实和是否开源系统关系不大。系统管理员需要关注的问题很多，包括存储、网络，也包括安全，有些公司的系统管理员甚至还要肩负应用程序上下线甚至交付工程方面的职责。因此，作为系统管理员需要不断地拓展自己的知识面，并思考如何能够使自己所管理的系统更有效地运行。我认为系统管理员除了普通的服务器软硬件方面的知识之外，还需要具备一定的开发能力，哪怕仅仅是把自己的一部分工作自动化地进行，因为这样做不仅可以提高工作效率、减少失误，而且可以让自己有更多的时间去学习和拓展知识面。

关于学习技巧，我个人认为小的 SA 团队，例如有 2-3 个人的小团队之间的相互促进效果是最明显的，在这种实际的工作中能够迅速提高能力并增长经验。

ChinaUnix: 能给我们谈谈最近几年的 Google SoC 么？FreeBSD 项目从中得到了什么发展？哪些公司或者团体对目前的 FreeBSD 项目贡献比较多？

delphij: Google SoC 是一个非常有远见的计划，我想我们应该感谢 Google 公司提供资助和持续不断地运营这个计划。过去四年 FreeBSD 从中受益很多，这不光是 SoC 所赞助的项目，也包括透过 SoC 所发现的人才，而今年我们继续作为 mentor organization 参加这个计划，这几年中已经完成

和正在进行的项目涵盖了操作系统的各个方面，从磁盘调度器、系统状态监控、托管访问控制、Xen、几种不同的文件系统等内核方面的实验性和应用性项目，到用户环境的流量监控、新的安装程序、更多的回归测试，以及采用 BSD 授权的一系列基础编译和调试工具等等，每年总共会有大约 20 个入选项目得到资助。

许多欧美和日本的公司对 FreeBSD 的开发提供赞助，或直接参与开发。例如大家熟悉的 Cisco、NetApp、Juniper 等等。除了资金赞助之外，还有很多公司是以捐赠代码或聘用 FreeBSD 开发人员的方式来支持 FreeBSD 开发的，除了 Google、Yahoo 和一些国外的硬件厂商之外，国内的网易、新浪两家网站都有直接聘用 FreeBSD 开发人员，这些对我们都是十分重要的支持。

ChinaUnix: 本来 Java 在 FreeBSD 平台的支持就磕磕碰碰的，现在好了，Oracle 收购了 Sun，FreeBSD 平台上的 Java 还会有好的发展么？

delphij: 我认为我需要修正一下，FreeBSD 对 Java 的支持并没有什么太严重的问题，过去 FreeBSD 对 Java 的支持不太好用的原因是 Sun 的授权不允许 FreeBSD 直接发布编译过的二进制包，这导致在 FreeBSD 平台上安装 Java 很不方便，因为编译一个二进制版本的包需要的时间比较长。

在 FreeBSD 基金会和 Sun 公司的努力下，Sun 公司授权 FreeBSD 基金会发布了一系列新版本的二进制 Java 版本，这在一定程度上缓解了先前由于必须自行编译或使用 Linux 兼容 ABI 支持使用 Linux 版本 Java 的不便。2006 年底，Sun 公司宣布以 GPLv2 发布一套名为 OpenJDK 的新的 Java 版本，并分步骤发布了较为完整的新版实现，FreeBSD 目前也提供了基于这组套件的新版本 Java，未来将取代 FreeBSD 现有的 native JDK。

Oracle 收购 Sun 目前仍在进行中，现在评估其影响可能还为时尚早。不过，由于 OpenJDK 已经以 GPLv2 发布了多个版本，Oracle 未来将其重新变为不公开源代码的可能性并不太大，最严重的情况也无非是新版不再公开代码而已，而根据美国法律，这并不影响社区继续维护已有版本的合法性，以及相关的研发能力。

相比 IBM，我认为 Oracle 收购 Sun 尽管不是十分理想的结果（个人感觉两家公司的企业文化差异还是比较大的），但是要比前者好一些，因为至少双方业务上的互补性更强一些。如果这次收购成功，也许能够给 Sun 这家领跑硅谷 27 年的技术公司注入新的活力。这次收购对于开源的影响，我想因为两家公司规模很大，还需要一段时间才能够看得比较清楚。

ChinaUnix: FreeBSD 平台的文件系统目前发展的如何，ZFS 可以在生产环境使用了么？相比较而言，Linux 平台的文件系统发展的很快，比如有最新的 Ext4、另外还有 Btrfs，能比较一下这几者间的差别么？

delphij: FreeBSD 目前提供了两种主要的文件系统——UFS 和 ZFS。在生产环境中使用早期的 ZFSv6 版本（直到 7.2-RELEASE 和之前的 FreeBSD 7.x 版本），在大量并发的条件下存在一些问题，而新的 ZFSv13 版本，也就是目前 7.2-STABLE 中的版本，则已经相当稳定，而且性能方面也有相当显著的改善。

ZFS 是一个非常关注数据可靠性的文件系统，正确配置和使用的 ZFS 与其他现有的文件系统，包括 UFS、ext3、ReiserFS、XFS 等等相比，都具有无可比拟的可靠性优势。当然，在性能方面，这些文件系统各自有不同的擅长的领域，例如 XFS 比较适合于跑数据库、ReiserFS 比较适合大量小文件的情形，然而正如 Donald E Knuth 说的，计算机硬件和程序都是“Garbage in, garbage out”的，好的程序员有一万种方法在不换文件系统的前提下改善性能，却没有任何一种办法在数据损毁的情况下让计算机得到正确结果。这一点上我非常赞同 Solaris 的一项设计原则：Performance is a

goal, correctness is a constraint.

ChinaUnix: 国内 BSD 系统部署的大概情况，主要应用在哪些方面，能给大家做个介绍么？

delphij: 我了解的国内部署主要有作为 CDN 的反向代理、国内一些知名网络游戏的服务器后端、商用的虚拟主机、DNS 等等。最近在美国这边也看到有报道说，国内的一些高校也基于 FreeBSD 开发了一些操作系统产品，并在一些国有银行中取代了 SCO Unix。FreeBSD 提供的防火墙和路由等机制非常适合于作为办公室、学校和网吧的安全设备。

ChinaUnix: FreeBSD 系统上的虚拟化技术支持目前进展如何？在这种技术趋势中 FreeBSD 能够扮演什么角色，另外还有云计算。

delphij: FreeBSD 从 4.0，也就是 2000 年左右的时候就已经提供了其特有的虚拟化技术——jail，这是一种轻量级的虚拟化技术，配合 FreeBSD 的 Linux ABI 支持，能够同时运行 FreeBSD 和 Linux 应用。与 Xen 相比，尽管它不能支持其他 OS 并行运行，但这种虚拟化的开销要少很多，能够满足许多大型互联网公司的需要，并且在过去近 10 年中被许多虚拟主机提供商使用。我们过去在公司内部的开发环境，以及一部分生产环境中也大批量地使用了 jail 技术，并达到了充分利用硬件资源的目的。

此外，最近一段时间 FreeBSD 的 Xen 支持也有相当多的进展（FreeBSD 目前已经能够很好地在 Xen 3.3 中运行，但 Amazon EC2 所采用的是较早版本的 Xen，因此有开发人员正在进行这方面的工作）。我认为支持 Xen（不仅是目前已经支持的 domU，也包括 dom0）对 FreeBSD 来说是一件很重要的事情，然而对普通用户，特别是希望尽量多地将硬件资源用于真正的业务和计算上的用户来说，jail 也许是个更好的选择，毕竟在真正的生产环境中需要同时混合 *nix 和 Windows 环境的机会并不是很多，类似 Xen 那样同时运行多个不同的内核的方法带来的开销就有些得不偿失了。

ChinaUnix: 能和大家谈谈 FreeBSD 操作系统的桌面支持进展如何？特别是硬件，比如显卡、声卡等等。

delphij: 这方面 FreeBSD 比较需要新的人手 :) 比较常见的硬件，如 Lenovo 和 Dell 的笔记本、常见的 AC97 声卡、Intel 3945/4965 无线网卡等，FreeBSD 已经能够提供相当完善的支持，但一些不太常见的硬件目前很可能还支持得不够好，或完全不支持。

对于希望参加 FreeBSD 团队的开发人员来说，这些都是很好的入手方向，因为驱动开发可以说是内核开发中最简单的部分。我们也希望硬件厂商能够尽可能地开放他们的硬件 datasheet，因为只开放二进制版本或缺少注释的源代码驱动出来，尽管也许能够使用，但非常不利于社区修正其中的 bug 和持续的维护，并且也会增加这些企业的人力成本。

ChinaUnix: FreeBSD 8.0 可能是下半年的重点吧，这个发行版将会带来什么新特性？

delphij: FreeBSD 8.0 目前已经进入了代码变动减速阶段，再过大约一个月之后将进入代码冻结，因此，基本上可以确定这个版本会在今年下半年发布（目前的计划是 8 月左右）。这个版本除了在先前版本基础上所作的性能和可靠性方面的改进之外，还有一些比较重要的改进：

- POSIX TTY。使 TTY 从驱动中抽象出来，并简化了两者的逻辑。有助于改善可靠性和 X 的用户体验。
- 节能。这是一项持续的改进计划，通过对系统中各个部分的重构来降低操作系统本身产生的能耗，并配合一些硬件特性来在运行过程中关闭暂时不用的硬件来节省电能。
- 轻量级内核线程。这个主要是帮助开发人员简化他们的设计模型。

- ECMP 和新的 ARP 实现。将流量在多个链路上路由的能力。
- DTrace。
- vimage 和 Xen 支持。对虚拟化的改进。
- NFSv4 支持。

原本我们还有计划在这个版本中正式引入基于 LLVM/Clang 的 C 编译器，实际上这部分的工作已经基本接近尾声了，但是由于代码变动减速阶段已经开始，我想这部分很可能会推迟到 8.0-RELEASE 正式发布之后再引入 9.0-CURRENT。

ChinaUnix: 最后一个问题，您做为资深的 BSD 用户，相信也用过 Linux，能简单的比较一下两者的差别么？比如代码质量、社区合作方式、商业支持等。

delphij: 从使用者的角度，FreeBSD 有一个叫 POLA 的规则，也就是说在新版本中尽可能减少用户可见的、不必要的界面变动。同样的配置，往往可以同时在不同的 FreeBSD 大版本升级之后继续使用。另一方面，FreeBSD 力求做到使用方式的统一，通过 ports/packages 套件安装的第三方软件，往往会修改成一致的、符合一定规范的安装方式，从而减少服务器维护的成本。

Linux 的开发模式更倾向于尽快地引入新的东西。我个人订阅了几种 BSD 操作系统、Linux 内核和 OpenSolaris 的提交邮件列表，从新增功能的引入就可以看出这一点。Linux 的开发方式中，会将改好的东西直接丢进 towards 的 git tree，这种方法有其优点——大家可以自己改自己的部分而不互相影响，但也有其缺点，对于开发人员来说，想要追踪代码的变动，或进行更大范围的同僚复审会比较困难，因为你要追踪的变动很可能在另一个 git 库中，甚至在作者自己的硬盘里面。我个人比较倾向于集中式的 SCM 管理，通过代码冻结这样的过程，能够让更多的人关注发行版本的品质，从而有助于产生更好的成果，然而，也有观点认为 FreeBSD 的开发模式对于新加入团队的开发者来说门槛过高，特别是同僚复审过程，有时对刚加入团队的开发人员来说，某种程度上会是心理上的挑战。

对于系统管理员来说，FreeBSD 提供了很多非常方便的性能诊断分析工具，例如 ktrace，能够帮助管理员了解系统正在进行的操作并迅速定位问题，而其预配置也多是基于安全和可靠性考虑。同时，由于系统中的不同部件风格非常一致，也有助于他们了解更多的技术和架构细节。

对于研发型用户（例如制作嵌入式设备的初创公司，等等）而言，FreeBSD 的优势在于代码风格统一以及宽松的授权。前者使得第三方开发人员能够很快地熟悉整个系统中各个部分的代码，减少在理解方面所遇到的阻碍，因为好的代码是具备良好的设计和实现，而不需要“情景分析”的。而宽松的授权则让这些用户能够在遵守授权规则的前提下，不需要担心新版本软件的授权突然将他们的行为列为违规而带来的法律风险；另外，更重要的是，*BSD 由于历史上所经历的法律诉讼，对代码的合法性审核非常严格，有问题的代码，哪怕仅仅是有可能侵犯专利，都会在第一时间被删除而不是成为日后被其他公司诉讼的隐患。在北美市场，由于这些因素的影响，已经有一些嵌入式厂商逐渐将产品转为使用 *BSD 和 FreeBSD 平台。

最近几年一家一直在支持和提供 FreeBSD 服务的公司——iXsystems（原 BSDi）在努力地拓展 FreeBSD 与硬件制造商之间的关系，并且在一些大用户的推动下取得了相当多的进展，而受其影响，在北美地区也有一些开发人员创业成立咨询公司来提供 FreeBSD 驱动移植等方面的服务，这些都会改善现有的一些驱动程序相对匮乏的现状。对于有兴趣开发 FreeBSD 硬件驱动的厂商，我们提供了一个名为 freebsd-drivers 的邮件列表来向他们提供技术支持。我个人也非常希望未来国内能够出现针对 FreeBSD 提供商业支持的公司。

本期推荐



调查说明:

为了更好地办好《开源时代》杂志,在本刊第10期(2009.07)发布之际,ChinaUnix《开源时代》编辑团队需要得到读者的反馈,在过去的9期中,你愿意和我们分享什么?我们重视您的宝贵意见!

《开源时代》致力于提供时效性与深度兼具的开源资讯和技术新知。并以独立立场和独到见解,为当今中国开源社区的中坚力量提供富有价值的业界洞察力!通过此调查问卷,我们将了解并发现本刊的不足,以便于在将来能够不断满足杂志读者的需求。

在线填写完整问卷,您将有机会获赠精美礼品!感谢您对《开源时代》的一贯支持!

奖品设置:

- 1、活动时间:2009年7月1日—2009年7月31日。
- 2、本次活动一共抽10个纪念奖,奖品均为价值50元的09ChinaUnix T恤5件和Linux相关图书5本。
- 3、以有效果问卷填写编号参加抽奖,每人不可重复获奖。
- 4、请如实填写相关资料以保证奖品的寄发,对虚假信息者与恶意投票者,不予抽奖。**凡正确填写信息且完成调查的用户,均可以获得ChinaUnix论坛200积分!**
- 5、通信与通讯方式不详的,在本网站公布1个月后仍没有取得有效联系,视为弃权。
- 6、开奖时间为调查结束一周之后。

在线调查:

在线调查地址: <http://linux.chinaunix.net/ebook/vote/>

如果您还有什么疑问或者对杂志的建议,欢迎和我取得联系:rmzhou@staff.chinaunix.net。

技术新知

MySQL Replication 优化的一些个人总结

ChinaUnix 网友: voxXu

这两天有空看了一下 ppc2009 大会的一些 pdf, 发现这次关于 Replication 的讨论还是很多的, 很多内容都很实用, 就总结了一下。本人英文水平有限, 难免有翻译的不对的地方, 有疑问欢迎讨论。

Replication 延时的类型

1、固定性的延时

——Slave 的数据持续性的落后于 Master 并且一直无法与 Master 的数据保持一致。

——Slave 的数据经常在白天落后于 Master, 而在晚上可以赶上并与 Master 的记录保持一致。

这种类型的延时通常是由于 Slave 服务器的负载已经到达了上限或在白天访问量大的时候到达上限造成的。

2、非固定性的延时

——Slave 的数据只是短暂的落后于 Master, 可在短时间内恢复, 这类型的延时通常与批量任务和报表有关, 效率差的查询也会导致这类延时。

Mysql Replication 的限制

Mysql 的 Replication 是单线程的, 意味着只能有效的使用一个 CPU 内核和一个磁盘, 一条复杂的查询或者事务都导致进程被阻塞, 不过现在针对 5.1 版本的多线程 Replication 补丁, <http://forge.mysql.com/wiki/ReplicationFeatures/ParallelSlave>, 还是 pre 版, 有很多限制, 感兴趣的可以去看看。

Replication 的容量

1、理解什么是 Replication 的容量

可以将 Replication 暂停一个小时, 重新启动 Replication 后, 观察 Slave 的数据多久可以与 Master 一致。从 Replication 重新启动到和 Master 数据一致所花费的时间与 Replication 暂停的时间的比值就是 Replication 的容量。

2、建议保持 Replication 的容量在 3 倍以上, 即延迟一个小时的数据, Slave 只需要 20 分钟就能与 Master 的数据一致。

Replication 的优化

1、5.0 的 mysql 中避免类似以下的更新语句

INSERT ... SELECT <complex query>

UPDATE WHERE <complex clause>

复杂的查询会导致 Replication 线程阻塞。如果是 insert 或 update 与 select 结合的语句, 可以讲 select 单独执行并保存在临时表中, 然后再执行 insert 或者 update。

如果使用的是 5.1 的 mysql, 新功能中的行级 Replication (RBR) 可以解决这个问题。RBR 可

以将在 Master 上通过复杂查询后更新的结果直接传给 Slave，Slave 可以直接将结果更新到数据库中。

2、避免大的事务

太大的事务会造成 Replication 长时间阻塞，数据会严重滞后于 Master。

Slave 服务器的硬件选择

更快的 CPU 内核，对于单线程的 Replication 多核 CPU 是没有任何优势的。更高速的硬盘，包括更高的转速和更好的高速缓存命中率，如果有钱的话上 SSD 吧。

主从结构的扩展性问题

1、如何降低写操作的频率

Master 的写操作会扩散到所有的 Slave 上，所以高频率的写操作会降低 Slave 的读操作效率。至少保持一台 Slave 做全库同步，其他的 Slave 可以只做部分表的同步。当然，这需要 web 应用程序的配合来分配哪些查询读哪些 Slave。将一些更新操作放到 memcached 中，例如 session 和计数器。Slave 使用 myisam 引擎，将一些写入量很大的更新操作直接在 slave 上执行，而不通过 Replication。

2、如何更有效的利用 Slave 的硬件资源

使用分区

有选择的对表进行同步

在 Slave 上对数据进行归档。

Session 的持久化

为不同的应用服务器分配不同的 Slave 进行读操作。

或者根据查询类型的不同来分配不同的 Slave。

3、如何使你的程序最大化的利用 Slave

将对数据更新不敏感的查询放到 Slave 上，而需要实时数据的查询则放到 Master。通过 session 的持久化，让做了修改的用户首先看到修改的内容，其他的用户可以等待 Slave 更新后再查看新内容。

对于某些数据，可以用 memcached 来存放数据的版本号，读 Slave 的程序可以先对比 Slave 的数据和 memcached 数据的版本，如果不一致则去读 master。用户和博客类的信息可以用这种方法。在查询前可以通过 SHOW SLAVE STATUS 检测 Slave 的状态，然后根据返回的结果进行服务器的选择。

软 RAID 部署系统分区之恢复攻略

ChinaUnix 网友: Jerrywj1

作为在几乎所有 Linux 发行版操作系统中所带的逻辑卷管理方式 (LVM)，其最大的特点是部署灵活和操作方便。而且在 Red Hat Enterprise Linux 中 LVM 也一直被作为默认的磁盘管理方式直接管理系统所在的设备文件。同时 LVM 能够很好地支持例如软件 Raid，裸设备等特殊磁盘类型，以及自带包括线性、条带、镜像、快照等多种功能，甚至在高版本系统中的 Lvm2 支持 RHCS (Red Hat Cluster Suite) 中的 HA LVM 和 Cluster LVM，能够实现 HA 集群中对块设备的资源共享。

写本文的初衷是因为最近不断在网上看到有朋友问及 RAID 尤其是软件 RAID 操作部署和管理方面的问题。其实网上这样的文章不在少数，但遗憾的是在个人看来大部分内容千篇一律，所涉及的实际应用和难度也差强人意，而且雷同者或者克隆者甚多。

比较突出的问题是，相信大多数人都知道怎么去做 raid，但是怎样维护，出了问题怎么去应用 Raid 功能的功能实现所谓的容错、备份和优化等，尤其是在一些关键的应用环境上，这方面内容则很少有人提及。

不是本人好出此风头，而是主要本着好人做到底，送佛送到西的原则，希望这篇文章能够对有需要的兄弟姐妹们提供一些帮助，当然各位在感激涕零的同时若能够良心发现考虑喝个茶，吃个包，增进一下感情的话，在下也乐意奉陪之至。

言归正传，这篇文章的目的是通过一个实际的案例来解释软件 RAID 的一些基本概念和特殊之处，同时关键的关键是在一些企业环境和某些业务上的应用以及维护管理方法。尽管在大多数情况下软件 Raid 给人的感觉是难登大雅之堂。

首先来说 RAID:

RAID 的概念对于所有的 administrators 来说应该都不陌生了。RAID 全称 “Redundant Array of Independent Disks” 即 “独立磁盘冗余阵列” 或简称 “磁盘阵列”。简单的说 RAID 是一种把多块独立的硬盘 (物理硬盘) 按不同的方式组合起来形成一个硬盘组 (逻辑硬盘) 从而提供比单个硬盘更高的存储性能和提供数据备份技术。组成磁盘阵列的不同方式称为 RAID 级别 (RAID Levels)。当然不同的级别对应了不同的功能。其中比较重要的当属冗余或者数据备份，数据备份的功能是在用户数据一旦发生损坏后，利用备份信息可以使损坏数据得以恢复，从而保障了用户数据的安全性。在用户看起来，组成的磁盘组就像是一个硬盘，用户可以对它进行分区，格式化等等。总之，对磁盘阵列的操作与单个硬盘一模一样。

在所有的 RAID 级别方面恐怕常用的就是 Raid-0, 1 和 5。对于这三种 Raid 级别的具体区别，我不想再花时间去解释，但是一定要注意的是软件 Raid 还是硬件 Raid。因为尽管在级别上的定义是一样的，但是其在实现方式上有着本质的区别，这才是本文要关心的核心问题。

所谓硬件 Raid 就是依靠主板和服务器上的 Raid 控制器 (硬件) 来实现的 Raid 功能；硬件 Raid 效率要比软件 Raid 更高，但是从操作上面来讲 Raid 的配置和控制都是在服务器硬件层 (BIOS) 设置中实现的。

多数的服务器在安装系统之前都必须要建立磁盘阵列，并且系统也只能安装在磁盘阵列上。那么一旦在 BIOS 完成这方面配置之后，无论是 Raid 的哪个级别，在操作系统中看到的磁盘都是最终形成的逻辑盘。

同样的，一旦 Raid 磁盘阵列出现问题，通常情况下修复也只能在硬件层面上进行。事实上这才

是我们所说的真 Raid，但是硬件 Raid 一般来讲都需要专门的 Raid 控制器实现，在价格方面自然是要高出不少。

在这种情况下就产生了一个所谓软件 Raid。Raid 实际上是依靠操作系统中自带的 Raid 配置软件来模拟硬件 Raid 实现阵列功能的一种做法。相对硬件 Raid 来说，软件 Raid 效率显得不太高，但是胜在价格方面有优势，尤其是在当今这种半死不活的经济环境之下尤为吸引。而且软件 Raid 还有一些其他的特性，比如其管理和控制可以在操作系统的层面上通过命令和工具实现；可以针对磁盘的分区来进行操作（尽管这是一个我现在都想不明白有什么用途的功能）。从性价比来看，所谓穷人的宝马就是这么来的。

综上所述，作为和硬件 Raid 最大的区别就是软件 Raid 从操作系统的角度中可以看到所连接的磁盘或者分区的数量，也能够通过 Raid 工具软件对其进行控制 and 操作。

不过各位不要高兴太早，因为在软件和硬件 Raid 之间，还有一种 Raid 称为 Host Raid，为什么说介乎于软硬之间呢？因为 Host Raid 是一种把初级的 RAID 功能附加给 SCSI 或者 SATA 卡而产生的产品，他是最早由 Intel 所推行的一套标准。实际上是把软件 RAID 功能集成到了产品的固件上，从而提高了产品的功能和容错能力。Host RAID 目前可以支持 RAID 0 和 RAID 1，而且归根结底还是软 Raid。一般现在很多 PC 机上板载的 Raid 控制器，多半就是这种。

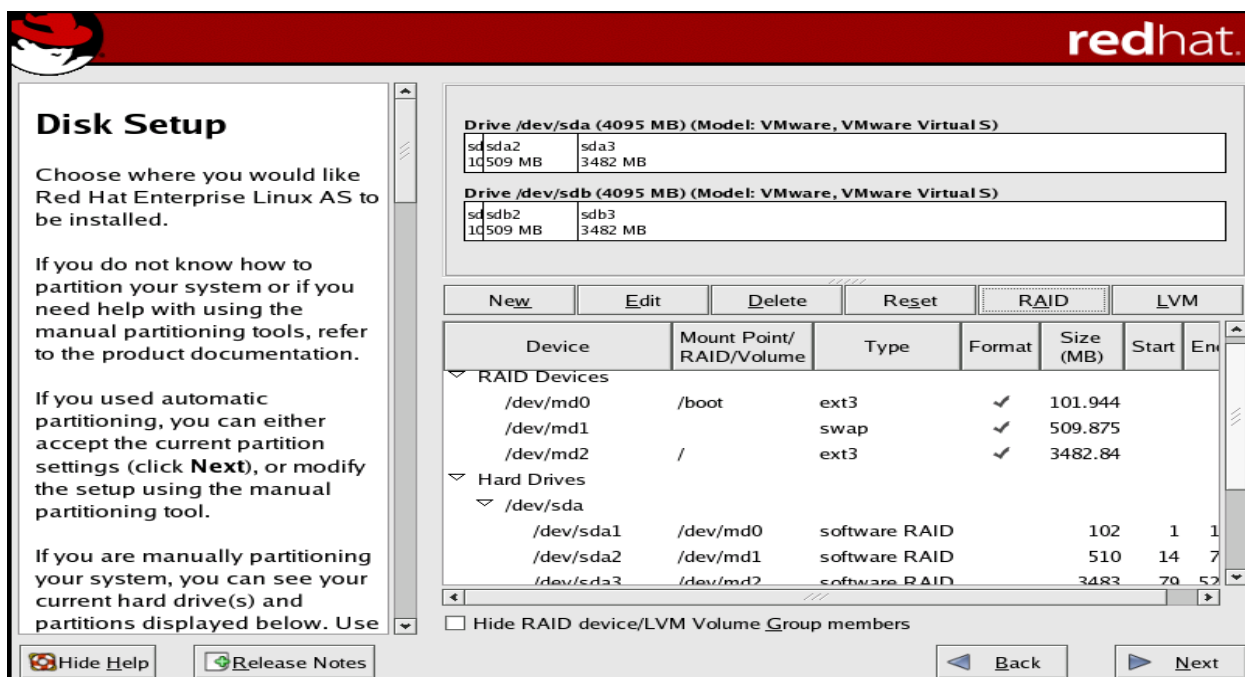
那么既然软件 Raid 要通过操作系统来控制 and 实现，因此有一个突出的问题就是他的数据同步只能同步到大多数情况下操作系统在正常运行中可见磁盘和分区的那个部分。也就是说，以 Raid-1 为例，如果所有系统分区都安装到 Raid-1 上，那么软件 Raid 同步的信息只能是第一分区开始到磁盘结束的那个部分，而在第一分区之前的一些内容，实际上两个磁盘是不同步的。简单地说就是系统安装中识别的第一块盘上会有启动引导记录等内容，而第二块盘上就没有了。这就产生一个麻烦的问题——如果我们将所有系统分区都安装在软 Raid-1 构建的磁盘阵列上，那么如果第二块盘损坏，系统仍然可以开机而这个磁盘阵列也能降级使用，但是如果恰恰损坏的是第一块盘，由于在第二块盘上没有启动引导信息而导致系统无法启动，自然这个时候再添加镜像什么的也将变得无比麻烦。

因此本文就是设计了这样一个场景，来以图文的方式演示将系统根以及启动分区都安装在软件 Raid-1 上，对于第一块硬盘损坏之后的解决方法。

对于学习而言，坚持给自己找点麻烦是十分必要的！不过对于这方面比较陌生的朋友来说最好找个虚拟环境进行测试，不要在玩出火的时候拎着一堆血淋淋的硬盘来戳我脊梁。

首先看我们的实验环境，一台 RHEL4U7 的虚拟机（选择最小安装），使用两块硬盘做 Raid-1，所有的系统分区都放在 Raid-1 上，包括/boot，swap 以及/分区。具体情况如下：

软 Raid-1 其实在分区的时候就已经确定和部署好了：



在系统安装完成之后的基本情况如下：

```
[root@localhost ~]# uname -r
```

2.6.9-78.ELsmp

```
[root@localhost ~]# fdisk -l
```

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	fd	Linux raid autodetect
/dev/sda2		14	78	522112+	fd	Linux raid autodetect
/dev/sda3		79	522	3566430	fd	Linux raid autodetect

Disk /dev/sdb: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------


```
/dev/sdb1 *      1      13   104391 fd Linux raid autodetect
/dev/sdb2      14      78   522112+ fd Linux raid autodetect
/dev/sdb3      79     522   3566430 fd Linux raid autodetect
```

Disk /dev/md0: 106 MB, 106823680 bytes

2 heads, 4 sectors/track, 26080 cylinders

Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md0 doesn't contain a valid partition table

Disk /dev/md2: 3651 MB, 3651928064 bytes

2 heads, 4 sectors/track, 891584 cylinders

Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md2 doesn't contain a valid partition table

Disk /dev/md1: 534 MB, 534577152 bytes

2 heads, 4 sectors/track, 130512 cylinders

Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md1 doesn't contain a valid partition table

默认挂载情况:

```
[root@localhost ~]# df -TH
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/md2	ext3	3.6G	806M	2.7G	24%	/
/dev/md0	ext3	104M	13M	86M	13%	/boot
none	tmpfs	131M	0	131M	0%	/dev/shm

以及/etc/fstab 中的内容:

```
[root@localhost ~]# cat /etc/fstab
```

```
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/md2      /          ext3 defaults    1 1
/dev/md0      /boot      ext3 defaults    1 2
none          /dev/pts   devpts gid=5,mode=620 0 0
none          /dev/shm   tmpfs defaults    0 0
none          /proc      proc  defaults    0 0
none          /sys       sysfs defaults    0 0
/dev/md1      swap       swap defaults    0 0
/dev/hdc      /media/cdrom auto pamconsole,exec,noauto,managed 0 0
/dev/fd0      /media/floppy auto pamconsole,exec,noauto,managed 0 0
```

Raid 阵列情况:

```
[root@localhost ~]# cat /proc/mdstat
```

Personalities : [raid1]

md1 : active raid1 sdb2[1] sda2[0]

522048 blocks [2/2] [UU]

md2 : active raid1 sdb3[1] sda3[0]

3566336 blocks [2/2] [UU]

md0 : active raid1 sdb1[1] sda1[0]

104320 blocks [2/2] [UU]

unused devices: <none>

```
[root@localhost ~]# mdadm -D /dev/md0
```

/dev/md0:

Version : 00.90.01

Creation Time : Fri May 1 18:45:44 2009

Raid Level : raid1

Array Size : 104320 (101.89 MiB 106.82 MB)

Device Size : 104320 (101.89 MiB 106.82 MB)

Raid Devices : 2

Total Devices : 2

Preferred Minor : 0

Persistence : Superblock is persistent

Update Time : Fri May 1 10:58:08 2009

State : clean

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

UUID : 03ca737f:77ff488e:5209468d:4ad4c3a7

Events : 0.44

Number	Major	Minor	RaidDevice	State
--------	-------	-------	------------	-------

0	8	1	0	active sync /dev/sda1
---	---	---	---	-----------------------

1	8	17	1	active sync /dev/sdb1
---	---	----	---	-----------------------

[root@localhost ~]# mdadm -D /dev/md1

/dev/md1:

Version : 00.90.01

Creation Time : Fri May 1 18:45:42 2009

Raid Level : raid1

Array Size : 522048 (509.90 MiB 534.58 MB)

Device Size : 522048 (509.90 MiB 534.58 MB)

Raid Devices : 2

Total Devices : 2

Preferred Minor : 1

Persistence : Superblock is persistent

Update Time : Fri May 1 10:54:49 2009

State : clean

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

UUID : 3444ec65:0a7fb6fd:d984f9e9:73c8ae0b

Events : 0.4

Number	Major	Minor	RaidDevice	State
--------	-------	-------	------------	-------

0	8	2	0	active sync /dev/sda2
---	---	---	---	-----------------------

1	8	18	1	active sync /dev/sdb2
---	---	----	---	-----------------------

[root@localhost ~]# mdadm -D /dev/md2

/dev/md2:

Version : 00.90.01

Creation Time : Fri May 1 18:45:43 2009

Raid Level : raid1

Array Size : 3566336 (3.40 GiB 3.65 GB)

Device Size : 3566336 (3.40 GiB 3.65 GB)

Raid Devices : 2

Total Devices : 2

Preferred Minor : 2

Persistence : Superblock is persistent

Update Time : Fri May 1 11:07:28 2009

State : clean

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

UUID : 9ea6a91b:43701808:c6a3f0ae:ed0e95cc

Events : 0.226

Number	Major	Minor	RaidDevice	State
0	8	3	0	active sync /dev/sda3
1	8	19	1	active sync /dev/sdb3

```
[root@localhost ~]# cat /etc/mdadm.conf
```

```
# mdadm.conf written out by anaconda
```

```
DEVICE partitions
```

```
MAILADDR root
```

```
ARRAY /dev/md2 level=raid1 num-devices=2  
UUID=9ea6a91b:43701808:c6a3f0ae:ed0e95cc
```

```
ARRAY /dev/md1 level=raid1 num-devices=2  
UUID=3444ec65:0a7fb6fd:d984f9e9:73c8ae0b
```

```
ARRAY /dev/md0 level=raid1 num-devices=2 UUID=03ca737f:77ff488e:5209468d:4ad4c3a7
```

现在开始动手破坏这个阵列，方法很简单，关机之后将第一块硬盘抽走就是了。我用的是虚拟机，所以这也就是动动手指头的事。

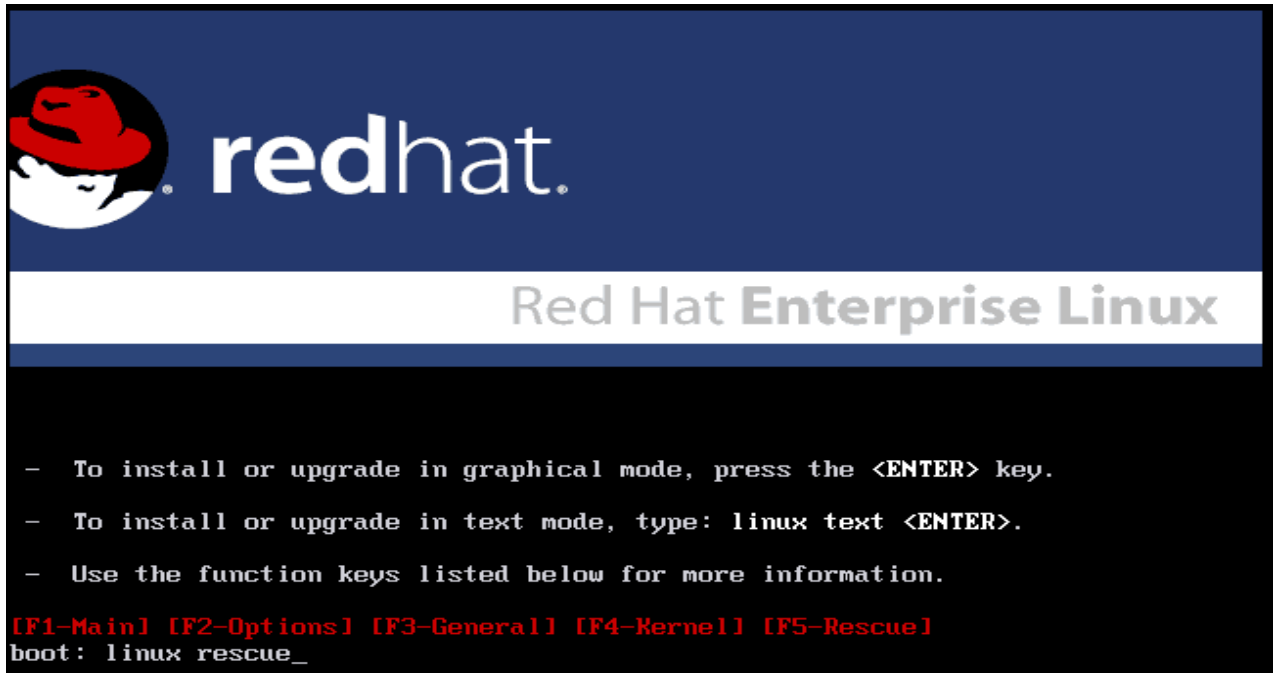
在开机之后的现象将是一片黑屏且没有任何信息，这显然说明软件 Raid-1 上无法对启动分区之前的内容进行全面的同步，至少开机连自检以及 GRUB 相关信息都没有，这样我们的破坏欲又一次得到了满足！

那么之后我们将有两种选择：

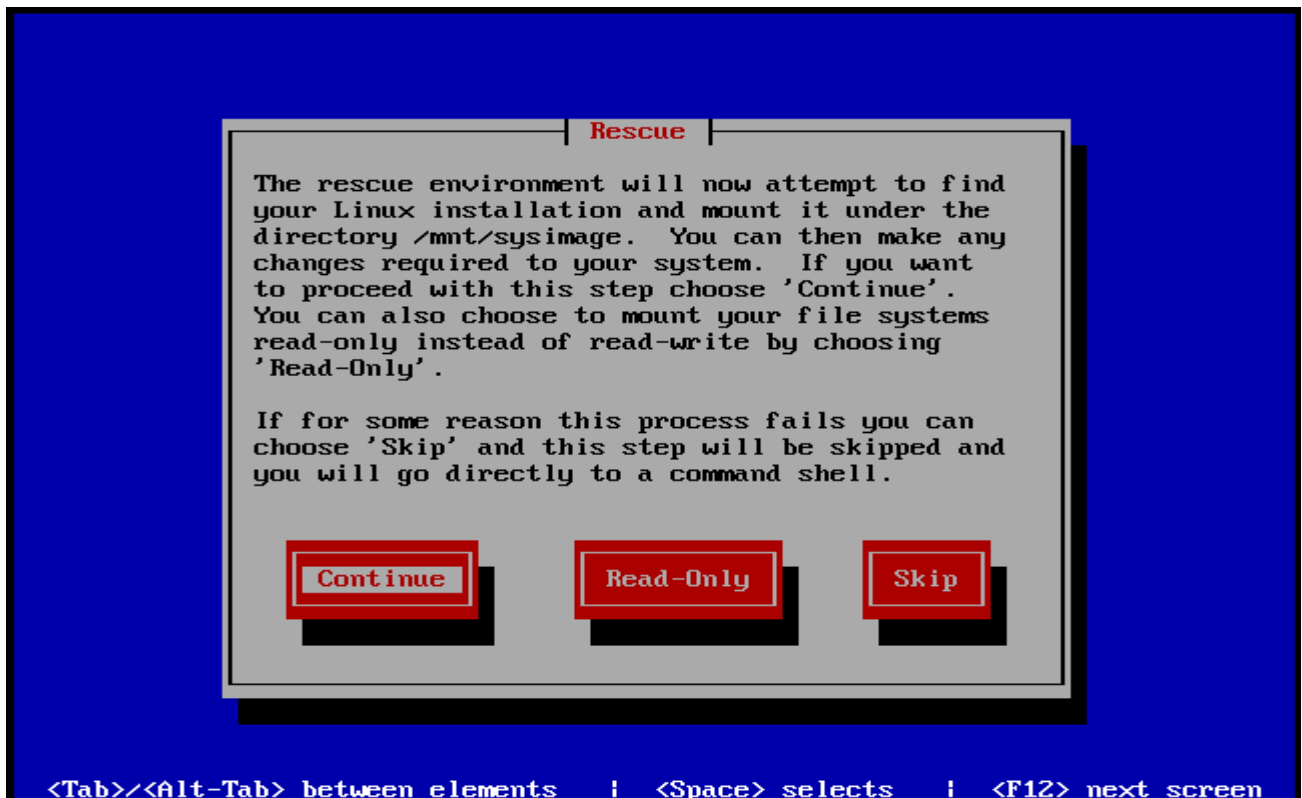
- 1、将原有的磁盘阵列降级为普通磁盘，至少将系统成功启动起来之后再去备份数据；
- 2、找一块新的盘替换拔走的盘以此来恢复原来的磁盘阵列以及其功能。

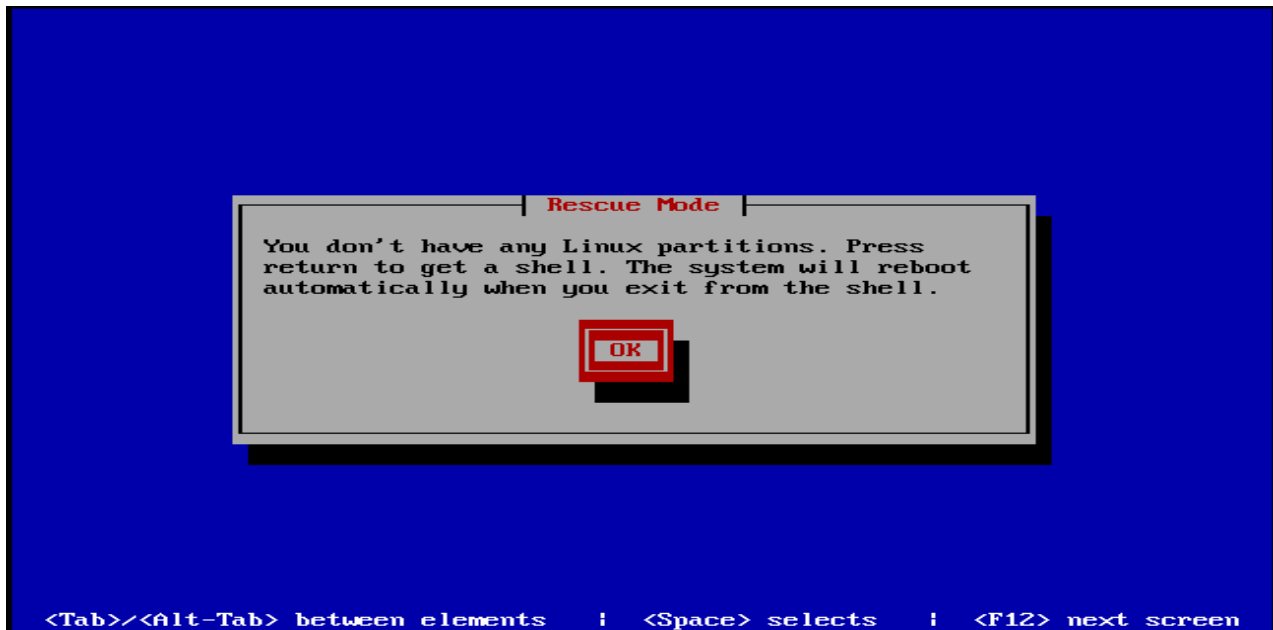
我们首先考虑第一种方法，即将原来的整个阵列降级为单个硬盘来启动系统。

在当前的状况下只能进入 rescue 模式了：



在 rescue 模式询问自动挂载根的时候我选择的是 continue，但是挂载是不成功的：





不要理会提示，进入系统之后，我们可以看到当前的分区状况，这个时候有一个关键步骤是将我的根所在分区挂载到/mnt/sysimage 目录下，当然/mnt/sysimage 要自己手工建立，因为一旦操作成功，我们将获得一个 grub 的操作环境（注意了操作和恢复方便，我将 swap 文件系统暂时禁用了）：

```
When finished please exit from the shell and your system will reboot.

-/bin/sh-3.00# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1  *           1           13       104391   fd  Linux raid autodetect
/dev/sda2             14          522      4088542+  fd  Linux raid autodetect
-/bin/sh-3.00# mkdir /mnt/sysimage
-/bin/sh-3.00# mount /dev/sda2 /mnt/sysimage/
-/bin/sh-3.00# grub
```

完成之后，即可运行 grub 命令来进入 grub shell，这样就可以手动将 GRUB 信息写入到 MBR 中去，我想如何在 grub shell 中安装 GRUB 应该不用我多说了，我以前有一篇关于系统启动 trouble shooting 的帖子曾经详细阐述过这个问题。操作完成之后 quit，退出 grub 的 shell。下面是截图：

```

lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]

grub> root (hd0,0)
  Filesystem type is ext2fs, partition type 0xfd

grub> kernel /vmlinuz-2.6.9-78.EL
  Possible files are: vmlinuz-2.6.9-78.EL vmlinuz-2.6.9-78.ELsmp

grub> kernel /vmlinuz-2.6.9-78.EL ro root=/dev/sda3
  [Linux-bzImage, setup=0x1e00, size=0x17754f]

grub> initrd /initrd-2.6.9-78.EL
  Possible files are: initrd-2.6.9-78.EL.img initrd-2.6.9-78.ELsmp.img

grub> initrd /initrd-2.6.9-78.ELsmp.img
  [Linux-initrd @ 0x32f000, 0xc0a83 bytes]

grub> setup (hd0)
  Checking if "/boot/grub/stage1" exists... no
  Checking if "/grub/stage1" exists... yes
  Checking if "/grub/stage2" exists... yes
  Checking if "/grub/e2fs_stage1_5" exists... yes
  Running "embed /grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
  Running "install /grub/stage1 (hd0) (hd0)1+16 p (hd0,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.

grub> quit_

```

事实上在完成这一个操作之后，系统可以直接重启，但这个时候系统是以一个降级 RAID 身份来运行的。

如图：

```

Personalities : [raid1]
md1 : active raid1 sda2[1]
      4088448 blocks [2/1] [_U]

md0 : active raid1 sda1[1]
      104320 blocks [2/1] [_U]

unused devices: <none>

```

这无疑将是一种简单高效和快捷的方法，尤其在阵列出现问题而导致系统无法启动的情况下，通过这样的操作能够给我们极大的信心！后续的操作其实不需要我多说了，可以直接将一个新盘插入到原来的槽位上，然后选择 mdadm /dev/md0 -a 命令向阵列中同步新盘。

但是如果将系统完全从一个阵列转换为单盘的话，显然事情还没有这么简单，因为即便开机 grub 加载成功，但是进入系统之后也因为原先的/etc/fstab 内容不正确而导致问题，所以需要手动将/etc/fstab 改正如下：

```

-/bin/sh-3.00# cat /mnt/sysimage/etc/fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/sda3          /                ext3      defaults        1 1
/dev/sda1          /boot            ext3      defaults        1 2
none               /dev/pts         devpts    gid=5,mode=620  0 0
none               /dev/shm         tmpfs     defaults        0 0
none               /proc            proc      defaults        0 0
none               /sys             sysfs     defaults        0 0
/dev/sda2          swap             swap      defaults        0 0
/dev/hdc           /media/cdrom     auto      pamconsole,exec,noauto,m
anged 0 0
/dev/fd0           /media/floppy    auto      pamconsole,exec,noauto,m
anged 0 0
-/bin/sh-3.00# mkdir /tempboot
-/bin/sh-3.00# mount /dev/sda1 /tempboot/

```


同时要改的还有启动分区的 grub.conf，那么我们可以临时创建一个空目录，将/dev/sda1 挂上：

```

-/bin/sh-3.00# cat /tempboot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/md2
#           initrd /initrd-version.img
#boot=/dev/sda1
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux AS (2.6.9-78.ELsmp)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.ELsmp ro root=/dev/sda3
    initrd /initrd-2.6.9-78.ELsmp.img
title Red Hat Enterprise Linux AS-up (2.6.9-78.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.EL ro root=/dev/sda3
    initrd /initrd-2.6.9-78.EL.img
-/bin/sh-3.00#

```

别以为就此万事大吉了，因为如果只改完这些之后就开机重启，系统会因为磁盘类型还是 software raid 而导致启动的时候加载 raid 模块出错并产生 kernel panic。所以最后的工作是 fdisk 将分区类型针对现有的情况分别改成 83 和 82，也就是标准 linux 磁盘分区以及 swap 分区：

```

-/bin/sh-3.00# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1  *           1           13       104391   fd  Linux raid autodetect
/dev/sda2             14           78      522112+   fd  Linux raid autodetect
/dev/sda3             79          522      3566430   fd  Linux raid autodetect
-/bin/sh-3.00# fdisk /dev/sda

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 83
Changed system type of partition 1 to 83 (Linux)

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 83
Changed system type of partition 3 to 83 (Linux)

Command (m for help): w_

```

同时将/etc/fstab 中的 mdadm.conf 信息全部注释：

```

When finished please exit from the shell and your system will reboot.

-/bin/sh-3.00# mkdir /mnt/sysimage
-/bin/sh-3.00# mount /dev/sda3 /mnt/sysimage/
-/bin/sh-3.00# cat /mnt/sysimage/etc/mdadm.conf

# mdadm.conf written out by anaconda
DEVICE partitions
MAILADDR root
ARRAY /dev/md2 level=raid1 num-devices=2 UUID=9ea6a91b:43701808:c6a3f0ae:ed0e95c
c
ARRAY /dev/md1 level=raid1 num-devices=2 UUID=3444ec65:0a7fb6fd:d984f9e9:73c8ae0
b
ARRAY /dev/md0 level=raid1 num-devices=2 UUID=03ca737f:77ff488e:5209468d:4ad4c3a
7
-/bin/sh-3.00#

```

因为改动分区类型不会影响文件系统及其上的数据信息，所以这两步操作可以放心执行。完成之后重新启动系统则一切正常：

```

md: autorun ...
md: ... autorun DONE.
Creating root device
umount /sys failed: 16
Mounting root filesystem
EXT3-fs: INFO: recovery required on readonly filesystem.
EXT3-fs: write access will be enabled during recovery.
kjournald starting. Commit interval 5 seconds
EXT3-fs: recovery complete.
EXT3-fs: mounted filesystem with ordered data mode.
Switching to new root
SELinux: Disabled at runtime.
INIT: version 2.85 booting
Setting default font (latarcyrheb-sun16): [ OK ]

Welcome to Red Hat Enterprise Linux AS
Press 'I' to enter interactive startup.
Setting clock (localtime): Fri May 1 11:45:47 CST 2009 [ OK ]
Starting udev: [ OK ]
Initializing hardware... storage network audio done [ OK ]
Configuring kernel parameters: [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname localhost.localdomain: [ OK ]
Your system appears to have shut down uncleanly
Press Y within 2 seconds to force file system integrity check..._

```

现在来看看第二种方法，即在抽走第一块硬盘之后添加一块新的硬盘来重组阵列并重启系统。应该来说，这才是一种彻底解决问题的方法。

不过在虚拟机里做这样的操作最大的问题是，当抽走原来的 sda 之后在原来的槽位上插入新盘，这块新盘将变为 sda，而原有的第二块硬盘会变成 /dev/sdb。但在虚拟机上，我没法像刚才那样在

/dev/sdb 上安装盒建立 grub 并指定系统去以/dev/sdb 启动，所以我这里采取一个简单的策略，即将/dev/sdb 上的第一个分区即启动分区以及整个盘头都 dd 到新的硬盘上，这样我至少能够确保系统能够启动起来。

当然在真实的环境中自然不用这样麻烦。因为完全可能在 BIOS 中指定以那块硬盘作为启动盘。

下面开始操作，针对新添加的硬盘，还是先进入 rescue 模式，按照原来的/dev/sdb 的结构进行分区。这里的分区需要严格按照柱面来划分确保和原来的/dev/sdb 一模一样。

```
Device Boot      Start   End  Blocks  Id System
/dev/sdb1 *         1    13    104391  fd  Linux raid autodetect
/dev/sdb2          14    78    522112+  fd  Linux raid autodetect
/dev/sdb3         79   522   3566430  fd  Linux raid autodetect
-/bin/sh-3.00# fdisk /dev/sda
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-522, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-522, default 522): 13

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (14-522, default 14):
```

这是完成之后的分区情况，将分区信息写入到分区表：

```

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1           13       104391   83   Linux
/dev/sda2           14           78      522112+   83   Linux
/dev/sda3           79          522     3566430   83   Linux

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): fd
Changed system type of partition 2 to fd (Linux raid autodetect)

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): fd
Changed system type of partition 3 to fd (Linux raid autodetect)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

-
```

现在将/dev/sdb1 和/dev/sdb 的盘头都 dd 到/dev/sda:

```

When finished please exit from the shell and your system will reboot.

-/bin/sh-3.00# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1           13       104391   fd   Linux raid autodetect
/dev/sda2           14           78      522112+   fd   Linux raid autodetect
/dev/sda3           79          522     3566430   fd   Linux raid autodetect

Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            *            1           13       104391   fd   Linux raid autodetect
/dev/sdb2           14           78      522112+   fd   Linux raid autodetect
/dev/sdb3           79          522     3566430   fd   Linux raid autodetect

-/bin/sh-3.00# dd if=/dev/sdb of=/dev/sda bs=512 count=1
1+0 records in
1+0 records out
-/bin/sh-3.00# dd if=/dev/sdb1 of=/dev/sda1
208782+0 records in
208782+0 records out
-/bin/sh-3.00#
```

因为盘头信息只需要第一个块，包含了 MBR 以及分区表，所以 512 字节足够。在 dd 完成之后像刚才那样将有文件系统的/dev/sdb3 挂载到/mnt/sysimage 目录下，这样既可获得 grub shell 进行后续的操作：

当然，为了确保万无一失，我恐怕需要针对现在的/dev/sdb 和/dev/sda 的盘头都要手动重写 GRUB 到 MBR 中，这样在/dev/sdb 中有启动系统的全部信息，而在/dev/sda 中有盘头到启动分

区的全部内容，这些信息对我们启动系统来说应该是足够了！

同时既然这次是有新的盘添加到原来的阵列中来进行 mirror，所以自然我不用再更改/etc/fstab 以及 grub.conf 文件，同时也不用改分区标志和/etc/mdadm.conf 文件：

```
-/bin/sh-3.00# mkdir /mnt/sysimage/
-/bin/sh-3.00# mount /dev/sdb3 /mnt/sysimage/
-/bin/sh-3.00# gr
grefer      grolbp      groupmod     grubby
grep        grolj4      groups       grub-install
grn         grops       grpck        grub-md5-crypt
grodvi      grotty      grpconv      grub-terminfo
groff       groupadd    grpunconv
groffer     groupdel    grub
-/bin/sh-3.00# grub
```

```
grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0xfd

grub> kernel /vmlinuz-2.6.9-78.EL
Possible files are: vmlinuz-2.6.9-78.EL vmlinuz-2.6.9-78.ELsmp

grub> kernel /vmlinuz-2.6.9-78.ELsmp ro root=/dev/md3
[Linux-bzImage, setup=0x1e00, size=0x168426]

grub> initrd /vmlinuz-2.6.9-78.EL
Possible files are: vmlinuz-2.6.9-78.EL vmlinuz-2.6.9-78.ELsmp

grub> initrd /initrd-2.6.9-78.EL
Possible files are: initrd-2.6.9-78.EL.img initrd-2.6.9-78.ELsmp.img

grub> initrd /initrd-2.6.9-78.ELsmp.img
[Linux-initrd @ 0x32f000, 0xc0a83 bytes]

grub> setup (hd0)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd0) (hd0)1+16 p (hd0,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.

grub>
```

```
grub> root (hd1,0)
Filesystem type is ext2fs, partition type 0xfd

grub> kernel /vmlinuz-2.6.9-78.EL
Possible files are: vmlinuz-2.6.9-78.EL vmlinuz-2.6.9-78.ELsmp

grub> kernel /vmlinuz-2.6.9-78.ELsmp ro root=/dev/sdb3
[Linux-bzImage, setup=0x1e00, size=0x168426]

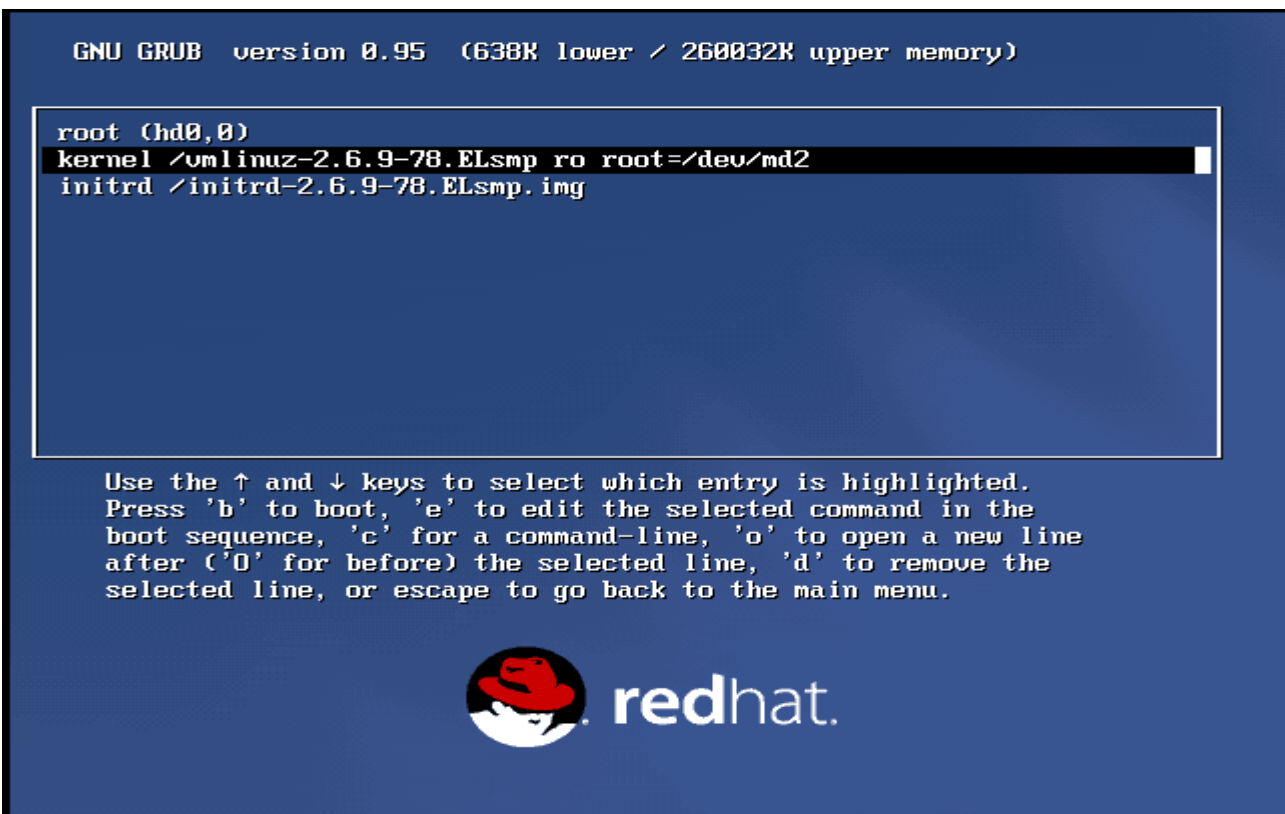
grub> initrd /initrd-2.6.9-78.EL
Possible files are: initrd-2.6.9-78.EL.img initrd-2.6.9-78.ELsmp.img

grub> initrd /initrd-2.6.9-78.ELsmp.img
[Linux-initrd 0 0x32f000, 0xc0a83 bytes]

grub> setup (hd1)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd1)"... 16 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd1) (hd1)1+16 p (hd1,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.

grub> _
```

完成之后 quit 退出 grub shell，之后 exit 重启系统，再次见到了我们熟悉的 GRUB 启动界面：



在系统启动过程中会有一个 fsck 的过程，但是由于我们事前没有进行过任何的阵列同步，所以现在整个的阵列是在降级使用中：

```

md: considering sdb1 ...
md: adding sdb1 ...
md: md0 already running, cannot run sdb1
md: export_rdev(sdb1)
md: ... autorun DONE.
Creating root device
Mounting root filesystem
kjournald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
Switching to new root
SELinux: Disabled at runtime.
INIT: version 2.85 booting
Setting default font (latarcyrheb-sun16):           [ OK ]

                Welcome to Red Hat Enterprise Linux AS
                Press 'I' to enter interactive startup.
Setting clock (localtime): Fri May 1 12:27:00 CST 2009 [ OK ]
Starting udev:                                     [ OK ]
Initializing hardware... storage network audio done [ OK ]
Configuring kernel parameters:                   [ OK ]
Loading default keymap (us):                      [ OK ]
Setting hostname localhost.localdomain:           [ OK ]
Checking root filesystem
/dev/md2 was not cleanly unmounted, check forced.
/dev/md2: i===== - 10.0%

```

通过 cat /proc/mdstat 或者 mdadm -D 都可以看到:

所以下面的工作就是分别手动同步这三个阵列: /dev/md0, /dev/md1 和/dev/md2

```

[root@localhost ~]# cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 sdb2[1]
      522048 blocks [2/1] [_U]

md2 : active raid1 sdb3[1]
      3566336 blocks [2/1] [_U]

md0 : active raid1 sda1[1]
      104320 blocks [2/1] [_U]

unused devices: <none>
[root@localhost ~]# mdadm /dev/md0 -a /dev/sdb1
RAID1 conf printout:
--- wd:1 rd:2
   disk 0, wo:1, o:1, dev:sdb1
   disk 1, wo:0, o:1, dev:sda1
mdadm: hot added /dev/sdb1
[root@localhost ~]# _

```

```

104320 blocks [2/1] [_U]

unused devices: <none>
[root@localhost ~]# mdadm /dev/md0 -a /dev/sdb1
RAID1 conf printout:
--- wd:1 rd:2
   disk 0, wo:1, o:1, dev:sdb1
   disk 1, wo:0, o:1, dev:sda1
mdadm: hot added /dev/sdb1
[root@localhost ~]# RAID1 conf printout:
--- wd:2 rd:2
   disk 0, wo:0, o:1, dev:sdb1
   disk 1, wo:0, o:1, dev:sda1

[root@localhost ~]# mdadm /dev/md1 -a /dev/sda2
RAID1 conf printout:
--- wd:1 rd:2
   disk 0, wo:1, o:1, dev:sda2
   disk 1, wo:0, o:1, dev:sdb2
mdadm: hot added /dev/sda2
[root@localhost ~]# RAID1 conf printout:
--- wd:2 rd:2
   disk 0, wo:0, o:1, dev:sda2
   disk 1, wo:0, o:1, dev:sdb2

```

在整个同步中，世间最漫长的是/dev/md2，因为这是/分区，可以通过 watch -n1 “cat /proc/mdstat” 查看到同步的状态和进程：

```

Every 1.0s: cat /proc/mdstat          Fri May  1 12:29:32 2009

Personalities : [raid1]
md1 : active raid1 sda2[0] sdb2[1]
      522048 blocks [2/2] [UU]

md2 : active raid1 sda3[2] sdb3[1]
      3566336 blocks [2/1] [_U]
      [====>.....] recovery = 20.6% (736640/3566336) finish=0.9min s
      peed=49109K/sec
md0 : active raid1 sdb1[0] sda1[1]
      104320 blocks [2/2] [UU]

unused devices: <none>

```

同步完成之后，整个阵列恢复也就成功了。若再重启系统，则没有任何手工干预的过程。实验到此成功！

总结：

通过上面的实验，我们完整地演示了一次软件 Raid 维护和修复的过程。可能在大多数情况下这套过程很多人都比较熟练。但是对于在系统根分区上部署软 Raid 的做法，恐怕使用的人就不怎么多了。

不过从上面的过程看，操作方法是一样的，只不过要涉及更多的系统启动原理，救援模式使用以及文件系统方面的内容。连同上期所发表的 LVM 缩根的文章，希望各位在铤而走险之前一定要先了解并且是比较透彻了解这些预备知识。毕竟实践出真知，但也尽量少走弯路。

VMware 环境下 OpenBSD 4.5 以及 Gnome 2.24 的安装

ChinaUnix 网友: OraBSD

Part I OpenBSD 4.5 的安装 (见 2009 年 5-6 期合刊)

Part II OpenBSD 环境下 Gnome 安装

声明: 本部分主要参考文献: atyu30.cublog.cn; 感谢 atyu30。

通过第一部分的学习, 相信大家对 OpenBSD 的基本操作已经了解, 其实和 Linux 都是相通的。

这部分我只是简单的介绍一下 Gnome 的安装。

※注: 要安装 Gnome, 最好在创建虚拟机的时候设置虚拟磁盘大于 6GB, /usr 分区 3GB 左右。虽然前面教程中的 /usr 2GB 分区能安装下 Gnome, 但是安装后改分区所剩空间不多, 如果只是做实验的话就没有什么。

设置 pkg_add 所需路径:

```
#export PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/amd64/
```

然后开始安装 Gnome:

```
#pkg_add -v gnome-desktop-2.24.3p3.tgz
```

```
#pkg_add -v gnome-session-2.24.3p4.tgz
```

为了让以后开机启动 Gnome Display Manager, 作如下设置:

```
#vi /etc/rc.local
```

```
#-----gdm-2.20.9p5-----
```

```
#To enable the GNOME Display Manger, add the following lines to the  
#/etc/rc.local file:
```

```
if [ -x /usr/local/sbin/gdm ];then
```

```
    echo -n 'gdm'; /usr/local/sbin/gdm
```

```
fi
```

编辑完后按 Esc 键, 再 shift+: wq Enter 保存并退出。

然后:

```
#echo "exec gnome-session" > ~/.xinitrc
```

最后, 执行 startx 进入 gnome!

```
#startx
```

因为 OpenBSD 自带了一个 fvwm2 轻量级桌面, 所以下面介绍一下 Gnome Desktop Manager” :



图 41

图 41 就是今后开机后的登陆界面，在 Username 下面的输入框里输入用户名，回车或者鼠标左键单击 OK 按钮确认，然后输入登陆密码，再确认就可以系统了。



图 42

如图 42 所示，鼠标左键单击 GNOME Desktop Manager 菜单栏的“Session”，会有一个下

来菜单，选择所要登陆的桌面环境:1.gnome desktop 就是选择进入 Gnome 桌面；2.Run Xclient script 就是选择进入 fvwm2 桌面。另外几个按钮就不一一介绍了。下面就两个截图吧：



图 43

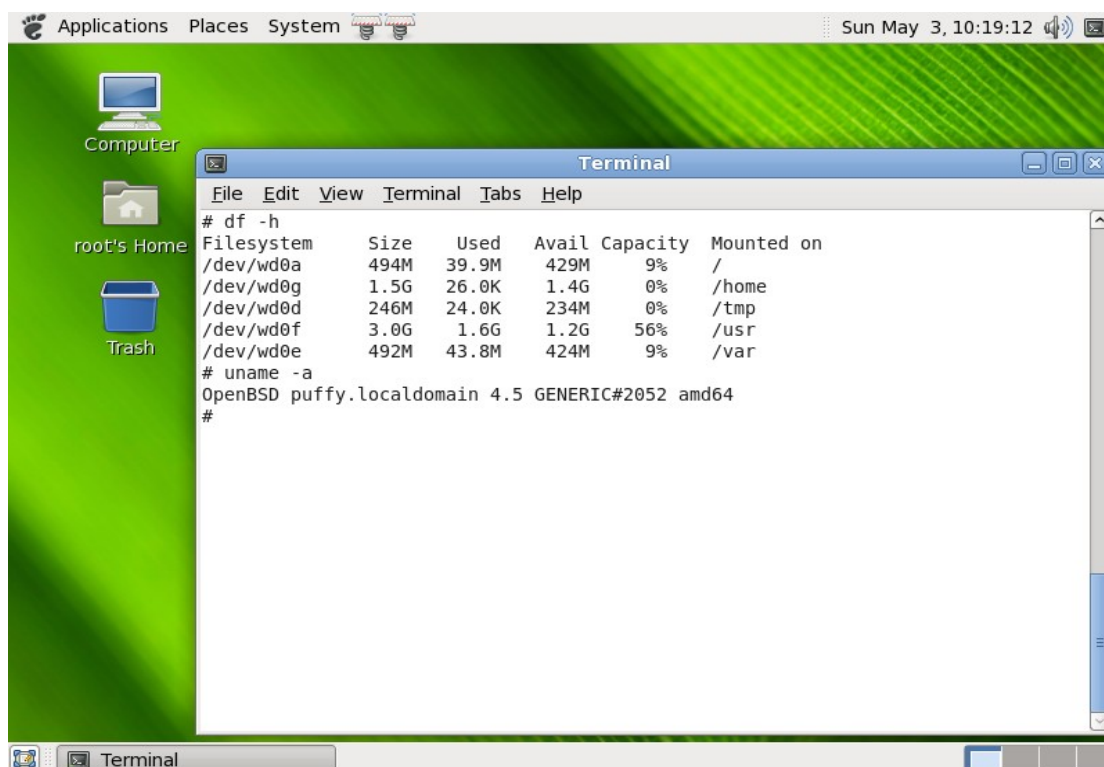


图 44

因为时间关系，上面两幅图是我在另一个虚拟机里面截取下来的。

说明一下：因为前面我说过，VMware 创建的虚拟机在硬盘上是碎片不但要对 Windows 系统做磁盘碎片整理，而且因为 VMware 本身的缺陷，还需要对 VMware 的虚拟硬盘作碎片整理（两处的碎片处理概念应该不一样，具体怎么解释我也说不清楚，对此我表示很抱歉。期待达人出来探讨）。总之，Windows 硬盘的碎片整理方法我就不说了，只说 VMware 虚拟机的碎片整理，方法如下：

打开 VMware 软件，虚拟机 OpenBSD 在关机状态下噢^_^

鼠标左键单击“Edit virtual machine settings”，然后会弹出一个对话框，再鼠标左键单击 Hardware 标签项下面的“Hard disk (IDE)”（我虚拟的硬盘类型是 IDE，显示字样视你的具体情况而定。）再鼠标左键单击右边的 Capacity 栏下面的“utilities”按钮，会有一个下拉菜单出现，再鼠标左键单击其中的“defragment”

整理成功后会有一个提示对话框，显示“Disk defragmentation successfully completed.”。这样做了以后在启动 OpenBSD，系统反应会比以前快多了，当然，快只是相对的，我还是觉得 gnome 在 OpenBSD 里反应很慢。而 RHEL 5 下的 gnome 反应都还是不错的，比起 OpenBSD 快多了。

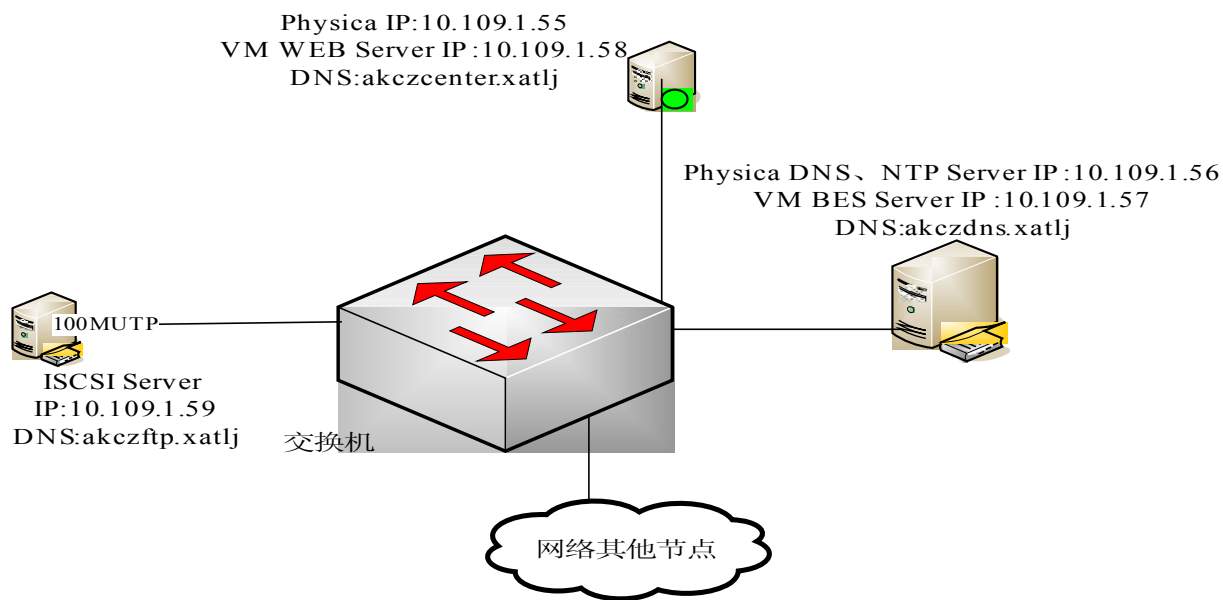
OK，到这里关于 OpenBSD 和 Gnome 的安装介绍就算是完成了。由于小弟是第一次做教程，囿于自身水平，不足之处还请大家多多批评指教。谢谢！

作者简介：原名谢昌均，chinaunix 论坛 ID 为 OraBSD，如果在 OpenBSD 方面还有什么问题的话，请通过电子邮件 BruceShea@gmail.com 与他取得联系。

Linux 集群系统 Vmware 虚拟机建设方案

ChinaUnix 网友：解宝琦

一、网络拓扑图及相关网络节点说明



机器名称	硬件 环境	IP	OS	作用和服务
Akczcenter.xatlj	华硕 TS 500 E5-PA4 标配	10.109.1.55	RHEL 5.1	● 集群节点，并安装 VMWARE，VM 创建在双机共享盘阵上，安装 WIN2000 系统，提供 WEB 服务。
Akczdns.xatlj	华硕 TS 500 E5-PA4 标配	10.109.1.56	RHEL 5.2	● 集群节点，并安装 VMWARE，提供 DNS NTP 服务，VM 创建在双机共享盘阵上，安装 WINXP 系统，提供 WEB 服务。
Akczftp.xatlj	华硕 TS 300 E5-PA4 标配	10.109.1.59	Fedora release 7	● ISCSI 服务器端，使用软 RAID 向集群节点提供 ISCSI 服务。本地 FTP 服务器。

二、服务器选型：

2. 1. iSCSI 服务器端，使用软 RAID 向集群节点提供 iSCSI 服务，本地 FTP 服务器。

服务器名称	iSCSI 服务器
主要功能	iSCSI 服务器端，使用软 RAID 向集群节点提供 iSCSI 服务。本地 FTP 服务器。
服务器型号	ASUS TS300E5-PA4
入选理由	作为 5U 入门级服务器提供快速处理能力和更高的性能。内建 RAID 0、1、0+1、5（软件受限于支持的操作系统）提供低成本的数据保护。此外，服务器级双千兆网卡提供专业可靠的网络连接。后置: 2 个 USB 2.0 接口可以使用 USB HUB 分两组连接 6 块 USB 移动硬盘，在操作系统支持下创建 RAID10 设备。
数量	1 台

2. 2. 集群节点，并安装 VMWARE，提供 DNS NTP 服务，VM1、VM2 创建在双机共享盘阵上，安装 WINXP 系统，提供 WEB 服务。

服务器名称	集群节点
主要功能	集群节点，并安装 VMWARE，提供 DNS NTP 服务，VM1、VM2 创建在双机共享盘阵上，安装 WINXP 系统，提供 WEB 服务。
服务器型号	ASUS TS300E5-PA4
入选理由	TS500-E4/PX4 服务器基于双核英特尔® 至强® 处理器 5000/5100 系列/四核英特尔® 至强® 处理器 5300 系列，支持 EM64T 技术、64 位操作系统、英特尔® VT 技术和英特尔® I/OAT 技术，经济实惠的完整数据保护
数量	2 台

三、iSCSI 服务器配置

该 iSCSI 服务器通过使用一台华硕 TS 300 E5-PA4 入门级的服务器以及 6 块移动硬盘，通过 LINUX 提供的软件 RAID 方法实现了一个大容量的、具有很强数据冗余能力以及很高数据安全性的数据存储案例。

3.1 双网卡绑定实现负载均衡

采用双网卡绑定技术实现服务器网络连接的高可用性及吞吐量。

1) 效果：网卡 eth0 和 eth1 虚拟成设备 bond0，并且向外公布一个虚拟 IP 地址。（拓扑图如图 1）

配置完成后在 akczftp:~ # 处输入命令：ifconfig，出现如下信息：

```
bond0  Link encap:Ethernet HWaddr 00:0C:76:6C:86:1F
        inet addr:10.109.1.59 Bcast:10.255.255.255 Mask:255.255.255.0
        inet6 addr: fe80::20c:76ff:fe6c:861f/64 Scope:Link
        UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
```

RX packets:10831 errors:0 dropped:0 overruns:0 frame:0
TX packets:476 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1290093 (1.2 Mb) TX bytes:164685 (160.8 Kb)

eth0 Link encap:Ethernet HWaddr 00:0C:76:6C:86:1F
inet addr:10.109.1.59 Bcast:10.255.255.255 Mask:255.255.255.0
inet6 addr: fe80::20c:76ff:fe6c:861f/64 Scope:Link
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:5396 errors:0 dropped:0 overruns:0 frame:0
TX packets:238 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:643234 (628.1 Kb) TX bytes:83262 (81.3 Kb)
Base address:0xb000 Memory:f5000000-f5020000

eth1 Link encap:Ethernet HWaddr 00:0C:76:6C:86:1F
inet addr:10.109.1.59 Bcast:10.255.255.255 Mask:255.255.255.0
inet6 addr: fe80::20c:76ff:fe6c:861f/64 Scope:Link
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:5435 errors:0 dropped:0 overruns:0 frame:0
TX packets:238 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:646859 (631.6 Kb) TX bytes:81423 (79.5 Kb)
Base address:0xc000 Memory:f9000000-f9020000

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:108 errors:0 dropped:0 overruns:0 frame:0
TX packets:108 errors:0 dropped:0 overruns:0 carrier:0

```
collisions:0 txqueuelen:0
```

```
RX bytes:7288 (7.1 Kb) TX bytes:7288 (7.1 Kb)
```

2) 配置方法：使用 VI 编辑器将标有//的内容添加到/etc/init.d/network 中的相应位置。（//不用输入，修改完后需要重启服务器操作系统）mode 为网卡工作方式，“1”为互援式工作方式（热备份工作方式，当前工作链路或网卡出现故障时，备份网卡会自动接管网络链路），“0”为集群式工作方式（同时在线提供两倍带宽，一条链路或网卡故障时，网络带宽为工作网卡带宽，不需切换）。如下：

```
akczftp:~ # vi /etc/network
```

```
akczftp:~ # vi /etc/init.d/network
```

```
        rc_status -v
        fi
```

```
//#modprobe bonding mode=active-backup miimon=100 primary=eth0
```

```
//modprobe bonding mode=0 miimon=100 primary=eth0
```

```
//ifconfig bond0 inet 10.109.1.59 netmask 255.255.255.0
```

```
//route add default gw 10.109.1.1
```

```
//ifenslave bond0 eth0
```

```
//ifenslave bond0 eth1
```

```
//ifconfig bond0 up
```

```
;;
```

```
stop)
```

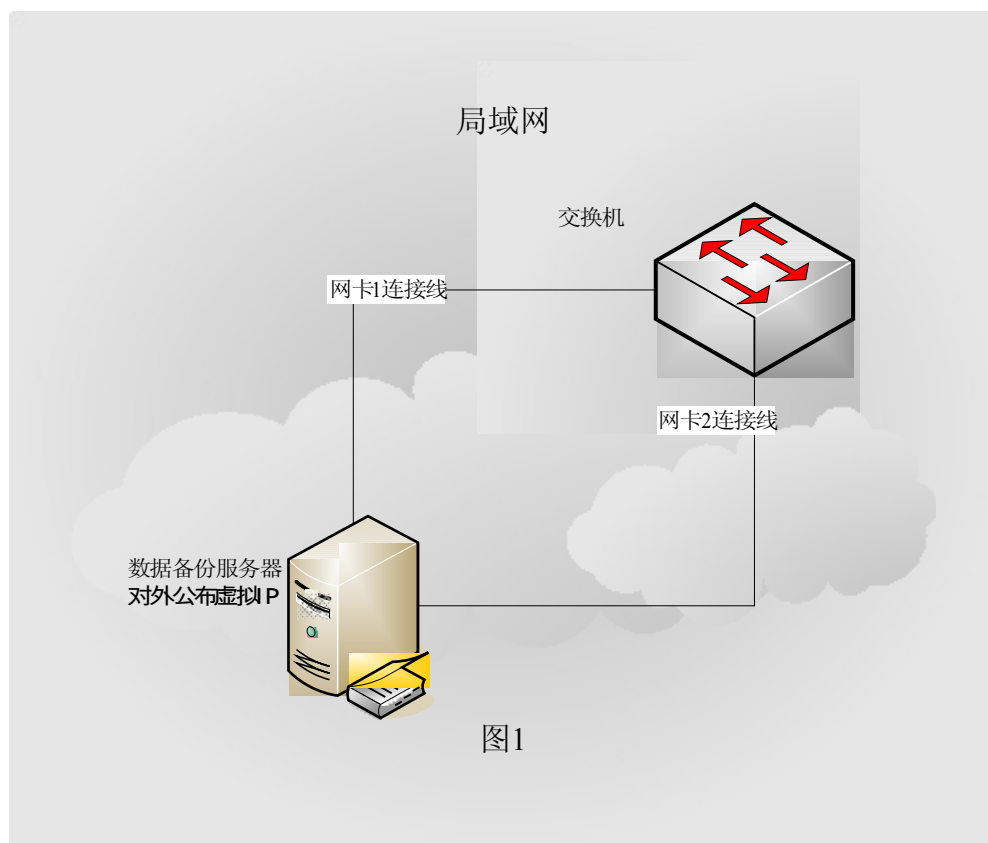
```
        echo Shutting down network interfaces:
```

```
echo Shutting down network interfaces:
```

```
//ifenslave -d bond0 eth0
```

```
//ifenslave -d bond0 eth1
```

```
//ifconfig bond0 down
```



3.2 Linux 创建软 Raid 的配置过程 (RAID1+0)

将 6 块廉价 IDE 硬盘盒组建成一个 RAID1+0 设备，操作系统启动正常后，分别依次开启 IDE 硬盘盒电源开关，操作系统会依次加载这些设备：

Disk /dev/sdb: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	9729	78148161	83	Linux

Disk /dev/sdc: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	9729	78148161	83	Linux

Disk /dev/sdd: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdd1		1	9729	78148161	83	Linux

Disk /dev/sde: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sde1		1	9729	78148161	83	Linux

Disk /dev/sdf: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdf1		1	9729	78148161	83	Linux

Disk /dev/sdg: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdg1		1	9729	78148161	83	

第一步：创建新磁盘分区：

用 root 用户执行命令 `# fdisk /dev/hdb`，完成对第二个 IDE 硬盘的格式化。

在 Command (m for help)：提示下输入 n，即创建新的磁盘分区。

在 Command action

e extended

p primary partition (1-4): 提示下输入 p, 即创建主分区。

在 Partition number (1-4): 提示下输入 1, 即分区号

在 First cylinder (1-522, default 1): 提示下按回车, 即指定起始柱面 (注意: 括号中尾柱面数字取决于硬盘的具体型号)

在 Last cylinder or +size or +sizeM or +sizeK (1-522, default 522): 提示下按回车, 即指定尾柱面。

在 Command (m for help): 提示下输入 p, 即查看分区状态, 如果看到 /dev/sdb1 等信息则说明分区成功。

在 Command (m for help): 提示下输入 t, 即转换分区类型

在 Hex code (type L list codes): 提示下输入 L, 即显示分区编码列表

在 Hex code (type L list codes): 提示下输入 fd, 即 LinuxRaid 分区编码

在 Command (m for help): 提示下输入 w, 即保存退出

重复以上步骤, 在其余两块硬盘上创建 LinuxRaid 分区。完成后需要重新启动 Linux 主机, 以使设置生效。

第二步: 创建 RAID 设备

执行创建 RAID1 设备命令

```
[root@localhost /]# mdadm -Cv /dev/md0 -l1 -n2 -c128 /dev/sd[b,c]1
```

```
[root@localhost /]# mdadm -Cv /dev/md1 -l1 -n2 -c128 /dev/sd[d,e]1
```

```
[root@localhost /]# mdadm -Cv /dev/md2 -l1 -n2 -c128 /dev/sd[f,g]1
```

命令中各参数的作用如下:

“-C” : 创建一个新的阵列; “/dev/md0” : 表示阵列设备名称; “-l1” 表示设置阵列模式, 可以选择 0、1、4、5、6, 它们分别对应于 RAID0、RAID1、RAID4、RAID5、RAID6, 这里设为 RAID1 模式; “-n2” 指设置阵列中活动设备的数目, 该数目加上备用设备的数目应等于阵列中的总设备数; “-c128” 指设置块的大小为 128KB, 缺省为 64KB; “/dev/hd[b, c,]1” 指当前阵列中包含的所有设备标识符, 也可以分开来写, 中间用空格分开。

当创建一个新阵列或者对阵列重构时, 设备需要进行同步操作。这一过程需要一定时间, 可以通过查看 /proc/mdstat 文件, 来显示阵列的当前状态以及同步进度、所需时间等信息。

```
[root@localhost etc]# cat /proc/mdstat
```

```
Personalities : [raid0] [raid1]
```

```
md3 : active raid0 md2[2] md1[1] md0[0]
```

```
234443904 blocks 128k chunks
```

```
md2 : active raid1 sgd1[1] sdf1[0]
```

```
78148096 blocks [2/2] [UU]
[=====>.....] resync = 42.5% (33272576/78148096) finish=88.1min
speed=8480K/sec
md1 : active raid1 sde1[1] sdd1[0]
78148096 blocks [2/2] [UU]
[=====>.....] resync = 43.6% (34123712/78148096) finish=85.6min
speed=8568K/sec
md0 : active raid1 sdc1[1] sdb1[0]
78148096 blocks [2/2] [UU]
[=====>.....] resync = 51.3% (40093760/78148096) finish=63.8min
speed=9933K/sec
unused devices: <none>
```

当新建或重构完成后，再次查看/proc/mdstat 文件:

```
[root@localhost ~]# cat /proc/mdstat
```

```
Personalities : [raid1] [raid0]
md2 : active raid1 sdg1[1] sdf1[0]
78148096 blocks [2/2] [UU]
md1 : active raid1 sde1[1] sdd1[0]
78148096 blocks [2/2] [UU]

md0 : active raid1 sdc1[1] sdb1[0]
78148096 blocks [2/2] [UU]
unused devices: <none>
```

通过以上内容，可以很清楚地看出当前阵列的状态，各部分所代表的意思如下：“[2/2]”中的第一位数表示阵列所包含的设备数，第二位数表示活动的设备数，如果有一个设备损坏，则第二位数将减1；“[UU]”标记当前阵列可以正常使用的设备情况，现假设/dev/md1 出现故障，则该标记将变成 [U_]，这时的阵列以降级模式运行，即该阵列仍然可用，但是不再具有任何冗余；

“sdd1[2]”指阵列所包含的设备数为 n，若方括号内的数值小于 n，则表示该设备为活动设备，若数值大于等于 n，则该设备为备份设备，当一个设备出现故障的时候，相应设备的方括号后将被标以(F)。

执行创建 RAID0 设备命令

```
[root@localhost etc]# mdadm -Cv /dev/md3 -l0 -n3 -c128 /dev/md0 /dev/md1 /dev/md2
```

查看/proc/mdstat 文件出现如下信息:

```
[root@localhost ~]# cat /proc/mdstat
```

Personalities : [raid1] [raid0]

md3 : active raid0 md2[2] md1[1] md0[0]

234443904 blocks 128k chunks

md2 : active raid1 sdg1[1] sdf1[0]

78148096 blocks [2/2] [UU]

md1 : active raid1 sde1[1] sdd1[0]

78148096 blocks [2/2] [UU]

md0 : active raid1 sdc1[1] sdb1[0]

78148096 blocks [2/2] [UU]

unused devices: <none>

至此，RAID10 盘阵创建完成。

第三步：配置 RAID 设备

接下来我们为 mdadm 生成配置文件，mdadm 的缺省配置文件为/etc/mdadm.conf，它主要是为了方便阵列的日常管理而设置的，对于阵列而言不是必须的，但是为了减少日后管理中不必要的麻烦，还是应该坚持把这一步做完。

在 mdadm.conf 文件中要包含两种类型的行：一种是以 DEVICE 开头的行，它指明在阵列中的设备列表；另一种是以 ARRAY 开头的行，它详细地说明了阵列的名称、模式、阵列中活动设备的数目以及设备的 UUID 号。格式如下：

```
DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
```

```
ARRAY /dev/md0 level=raid5 num-devices=3 UUID=8f128343:715a42df:baece2a8:a5b878e0
```

以上的这些信息可以通过扫描系统的阵列来获取，命令为：

```
# mdadm -Ds
```

```
ARRAY /dev/md0 level=raid1 num-devices=2  
UUID=c4f17506:687b1bd2:9964e709:c0d4630d
```

```
ARRAY /dev/md1 level=raid1 num-devices=2  
UUID=ae52d48a:e65c88c2:0bb4c763:50499806
```

```
ARRAY /dev/md2 level=raid1 num-devices=2 UUID=4027a9e7:a8fab70b:272f6793:c8a72f12
```

```
ARRAY /dev/md3 level=raid0 num-devices=3  
UUID=1cdfe4da:d88fe61e:54d2ed07:7aeb9c96
```

使用 vi 命令，按照规定的格式编辑修改/etc/mdadm.conf 文件

```
# vi /etc/mdadm.conf
```

```
[root@localhost etc]# vi /etc/mdadm.conf
```

文件内容如下：

```
ARRAY /dev/md0 level=raid1 num-devices=2
UUID=c4f17506:687b1bd2:9964e709:c0d4630d devices=/dev/sdb1,/dev/sdc1

ARRAY /dev/md1 level=raid1 num-devices=2
UUID=ae52d48a:e65c88c2:0bb4c763:50499806 devices=/dev/sdd1,/dev/sde1

ARRAY /dev/md2 level=raid1 num-devices=2 UUID=4027a9e7:a8fab70b:272f6793:c8a72f12
devices=/dev/sdf1,/dev/sdg1

ARRAY /dev/md3 level=raid0 num-devices=3
UUID=1cdfe4da:d88fe61e:54d2ed07:7aeb9c96 devices=/dev/md0,/dev/md1,/dev/md2
```

//每行的 devices 后的内容为添加内容，设备名以盘阵的设备名为准。

重启服务器后，按照顺序依次打开硬盘盒，执行如下命令：

```
[root@localhost etc]# mdadm -As
```

出现如下信息：

```
mdadm: /dev/md0 has been started with 2 drives.
mdadm: /dev/md1 has been started with 2 drives.
mdadm: /dev/md2 has been started with 2 drives.
mdadm: /dev/md3 has been started with 3 drives.
```

说明系统已经挂载完成创建好的 RAID10 阵列。

查看 /proc/mdstat 文件出现如下信息：

```
[root@localhost ~]# cat /proc/mdstat

Personalities : [raid1] [raid0]

md3 : active raid0 md0[0] md2[2] md1[1]
      234443904 blocks 128k chunks

md2 : active raid1 sdf1[0] sdg1[1]
      78148096 blocks [2/2] [UU]

md1 : active raid1 sdd1[0] sde1[1]
      78148096 blocks [2/2] [UU]

md0 : active raid1 sdb1[0] sdc1[1]
      78148096 blocks [2/2] [UU]

unused devices: <none>
```


第四步：创建 LVM 卷设备

在[root@akczftp ~]# 输入 pvcreate /dev/md3 命令，创建物理卷。

在[root@akczftp ~]# 输入 vgcreate akczftp /dev/md3 命令创建卷组。

分别在[root@akczftp ~]# 输入 lvcreate -L 120000M -n akczftpLV1 akczftp、lvcreate -L 100000M -n akczftpLV2 akczftp 两个命令创建两个逻辑卷。

使用 VI 编辑器在/etc/init.d/文件中创建启动和停止 iscsi 服务的脚本：

启动脚本/etc/init.d/mdmountvsftpd_start 内容如下：

```
#!/bin/bash
mdadm -As
vgchange -a y akczftp
mount /dev/dm-0 /ftp （作为本地目录使用）
/etc/init.d/vsftpd start
/etc/init.d/iscsi-target start
```

停用脚本/etc/init.d/mdmountvsftpd_stop 内容如下：

```
#!/bin/bash
/etc/init.d/iscsi-target stop
umount /ftp
vgchange -a n akczftp
mdadm --stop /dev/md3
mdadm --stop /dev/md2
mdadm --stop /dev/md1
mdadm --stop /dev/md0
```

注意：

系统开机顺序：

- 启动操作系统
- 操作系统正常启动后依次打开硬盘盒电源

3.3 在此基础上我们将安装 Linuxlscsi 服务器需要的服务器端软件 iscsitarget，该软件可以从 <http://iscsitarget.sourceforge.net/> 进行下载，下载完后在本地下载保存目录中应该出现 iscsitarget-0.4.16.tar.gz 文件，注意：安装要 2.6.14 以上内核。在[root@akczftp opt]# uname -a

系统返回如下信息：

```
Linux akczftp.xatlj 2.6.21-1.3194.fc7 #1 SMP Wed May 23 22:35:01 EDT 2007 i686 i686 i386
```

GNU/Linux

2.6.21-1.3194.fc7 即为该系统 Linux 内核 (Kernel) 版本。

iscsitarget 具体安装步逐如下:

3.3.1 在[root@akczftp opt]#处输入命令 tar -zxf iscsitarget-0.4.15.tar.gz //解压该 tar 包

3.3.2 在[root@akczftp opt]#处输入命令 cd iscsitarget-0.4.16 //进入解压后生成的安装文件目录

3.3.3 依次在[root@akczftp iscsitarget-0.4.16]#处输入以下命令

make

make kernel

make usr

make KSRC=/usr/src/kernels/2.6.21-1.3194.fc7-i686/ install-man

make KSRC=/usr/src/kernels/2.6.21-1.3194.fc7-i686/ install-kernel

make KSRC=/usr/src/kernels/2.6.21-1.3194.fc7-i686/ install-usr

make KSRC=/usr/src/kernels/2.6.21-1.3194.fc7-i686/ install-init

make KSRC=/usr/src/kernels/2.6.21-1.3194.fc7-i686/ install-etc

chkconfig --add iscsi-target

3.3.4 在[root@akczftp opt]#处输入命令 vi /etc/ietd.conf 对/etc/ietd.conf 文件进行配置, 以下内容是我的配置实例

Target iqn.2008-04.xatlj.akczftp:storage.disk2.sys1.xyz

Users, who can access this target. The same rules as for discovery

users apply here.

Leave them alone if you don't want to use authentication.

#IncomingUser joe secret

#OutgoingUser jim 12charpasswd

IncomingUser admin akczadmin

OutgoingUser admin akczadmin

Logical Unit definition

You must define one logical unit at least.

Block devices, regular files, LVM, and RAID can be offered

to the initiators as a block device.

Lun 0 Path=/dev/dm-1,Type=fileio

Alias iDISK0

```
# Alias Test

# various iSCSI parameters

# (not all are used right now, see also iSCSI spec for details)

#MaxConnections      1
#InitialR2T           Yes
#ImmediateData        No
#MaxRecvDataSegmentLength 8192
#MaxXmitDataSegmentLength 8192
#MaxBurstLength       262144
#FirstBurstLength     65536
#DefaultTime2Wait     2
#DefaultTime2Retain   20
#MaxOutstandingR2T    8
#DataPDUIInOrder      Yes
#DataSequenceInOrder Yes
#ErrorRecoveryLevel   0
#HeaderDigest         CRC32C,None
#DataDigest           CRC32C,None
# various target parameters
#Wthreads             8
```

上面这一行中的 “Target iqn.2008-04.xatlj.akczftp:storage.disk2.sys1.xyz

” 这是 Target 名称，称作 ‘iSCSI Qualified Name’ (简称: iqn)，这格式在 iSCSI 规格书中有详细说明：

iqn.yyyy-mm.<reversed domain name>[:identifier]

格式 意义 范例

yyyy-mm 年份-月份 2008-04

reversed domain name 倒过来的网域名称 xatlj.akczftp

identifier 识别代号 storage.disk2.sys1.xyz

LUN (Logical Unit Number)， “#Lun 0 Path=/dev/dm-1,Type=fileio” 这一行中，Path 改成您提供的硬盘名称，此例为 “Path=/dev/dm-1”。我们可以使用像是区块装置 (Block devices, 例 /dev/sdc)、一般档案 (regular files)、LVM 逻辑卷轴 (Logical Volume) 与磁碟阵列 (RAID) 提供给 Initiators 主机当做 ‘区块装置’ (硬盘)。

3.3.5 在[root@akczftp etc]#处输入/etc/init.d/iscsi-target start 启动 iscsi-target 服务。

3.3.6 在[root@akczftp etc]# 处输入 cat /proc/net/iet/volume 命令检查启动后的 iscsi-target, 如果系统返回如下信息

```
tid:1 name:iqn.2008-04.xatlj.akczftp:storage.disk2.sys1.xyz
```

```
lun:0 state:0 iotype:fileio iomode:wt path:/dev/md3
```

说明 iscsi-target 服务已经正常工作。

四、RHEL5 Cluster 部署

4、1 按要求分别在两台服务器上安装 RHEL5.1 和 RHEL5.2 操作系统及 Cluster 需要的软件环境。

4、2 配置 iscsi 客户端及共享存储

在 RHEL5.1 系统中, 生成并查看 lscsi 启动器的名称

在[root@akczdns etc]#处输入命令

```
echo "InitiatorName=`iscsi-iname`" > /etc/iscsi/initiatorname.iscsi
```

在[root@akczdns etc]#处输入命令

cat /etc/iscsi/initiatorname.iscsi, 系统返回如下信息

```
InitiatorName=iqn.2005-03.com.redhat:01.9fd8e65dd7f8
```

在 RHEL5.1 系统中, 配置 iSCSI 启动器服务

在[root@akczdns etc]#处输入命令 vi /etc/iscsi/iscsid.conf (iSCSI 启动器服务的配置文件,按照具体情况修改相关行内容)

```
node.startup = automatic
```

```
node.session.auth.authmethod = CHAP
```

```
node.session.auth.username = iqn.2005-03.com.redhat:01.9fd8e65dd7f8
```

```
node.session.auth.password = 01.9fd8e65dd7f8
```

在[root@akczdns etc]#处输入命令 chkconfig iscsi --level 35 on

在 Storbridge 系统中, 创建并分配一个 iSCSI 共享

通过共享管理-iSCSI 共享, 使用 iSCSI 共享虚拟磁盘创建一个 iSCSI 共享;

根据第 3 步得到的 iSCSI 启动器的名称, 使用 CHAP 认证模式进行分配;

启动器名称: iqn.2005-03.com.redhat:01.9fd8e65dd7f8

启动器口令: 01.9fd8e65dd7f8

在 RHEL5 系统中, 启动 iSCSI 启动器服务

在[root@akczdns etc]#处输入命令 service iscsi start

在 RHEL5 系统中, 使用 iSCSI 存储

在[root@akczdns etc]#处输入命令 rm -rf /var/lib/iscsi/nodes/*

在[root@akczdns etc]#处输入命令 rm -rf /var/lib/iscsi/send_targets/*

在[root@akczdns etc]#处输入命令 iscsiadm -m discovery -t st -p 10.109.1.59 (发现)

```
10.109.1.59:3260,1 iqn.2008-04.xatlj.akczftp:storage.disk2.sys1.xyz
```

在[root@akczdns etc]#处输入命令 iscsiadm -m node -T

```
iqn.2001-04.xatlj.akczftp:storage.disk2.sys1.xyz -p 10.109.1.59 -l (登录)
```

在[root@akczdns etc]#处输入命令 sfdisk -s

```
/dev/rd/c0d0: 71802880
```

```
/dev/sda: 102400000
```

```
total: 174202880 blocks
```

其中/dev/sda 就是我们的 lscsi 存储设备

在 RHEL5.2 系统中同样执行以上步逐，正常情况可以看到如下信息：

```
[root@akczcenter ~]# sfdisk -s
```

```
/dev/rd/c0d0: 137498624
```

```
/dev/sda: 102400000
```

```
total: 239898624 blocks
```

这样我们可以在两台机器上看到相同的盘符信息。

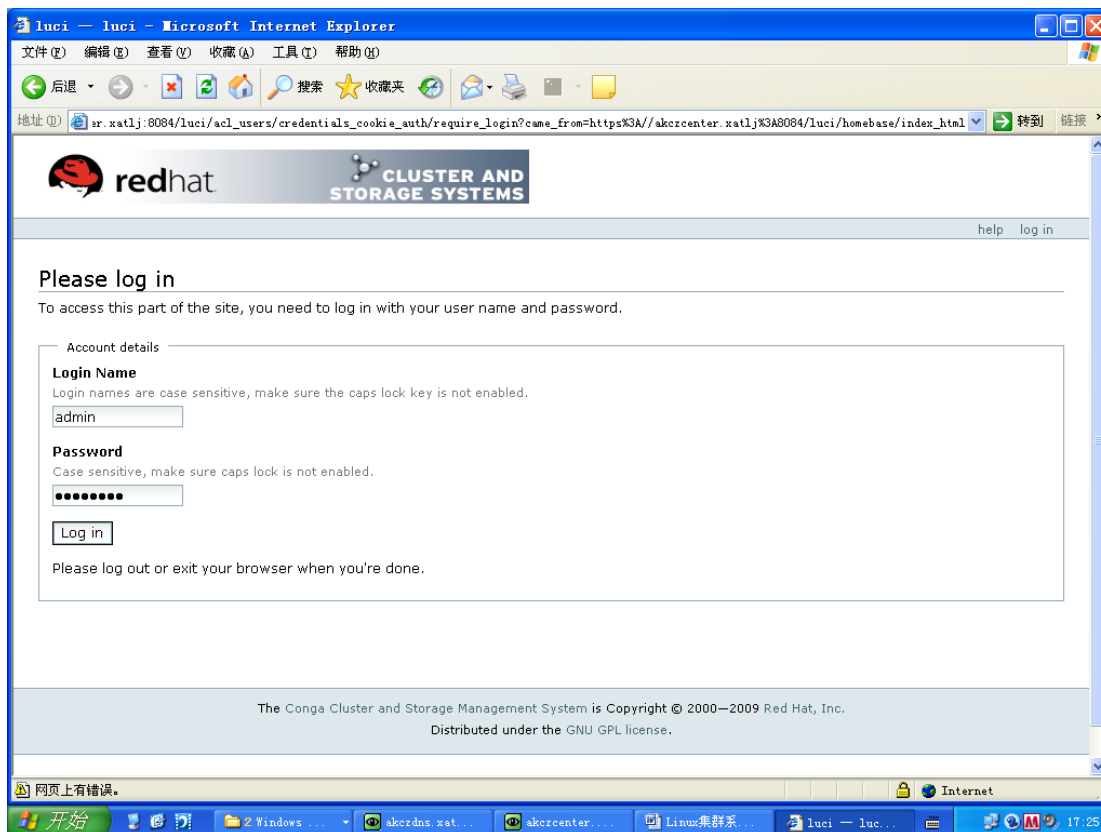
我们分别在两台机器中执行如下命令：

mkfs.gfs2 -t akczcluster:akczftp -p lock_dlm -j 2 /dev/sda 将/dev/sda 格式化为 DLM 锁方式 GFS2 文件系统。

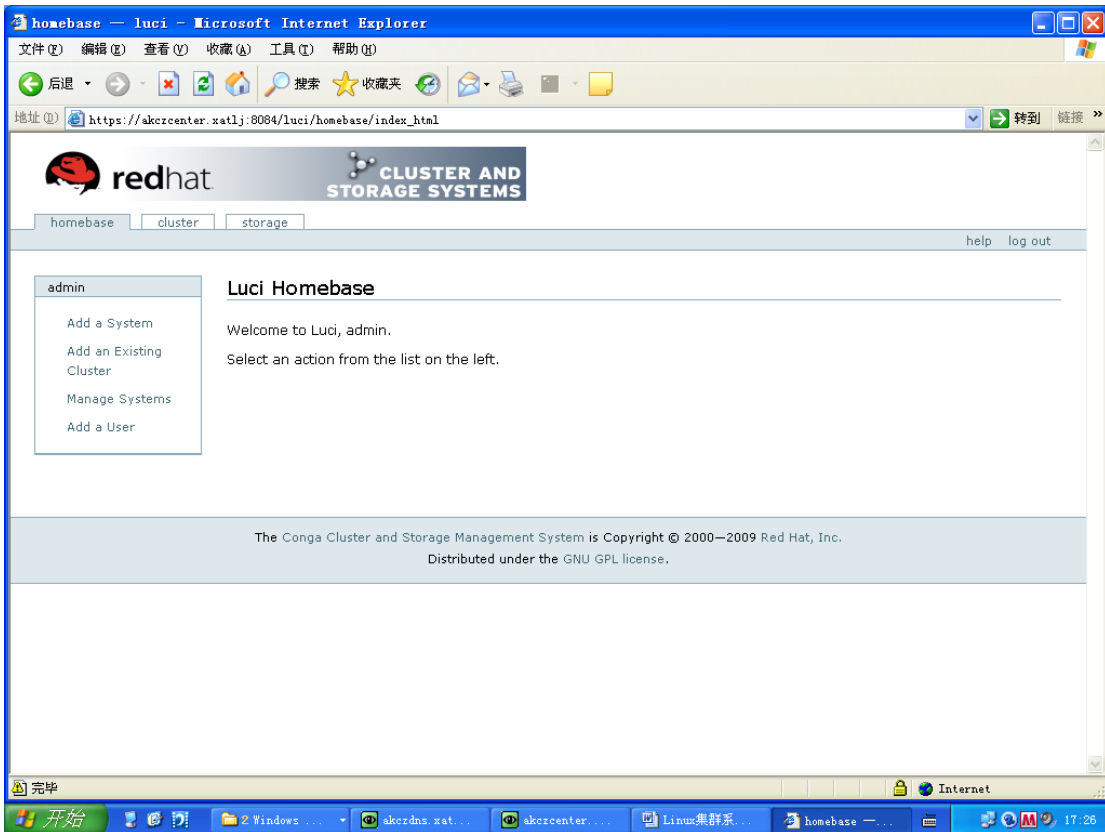
4. 3 创建 Cluster

在 IE 浏览器中输入

https://akczcenter.xatlj:8084/luci/acl_users/credentials_cookie_auth/require_login?came_from=https%3A//akczcenter.xatlj%3A8084/luci/homebase/index_html 出现如图画面，输入管理员帐号密码登陆

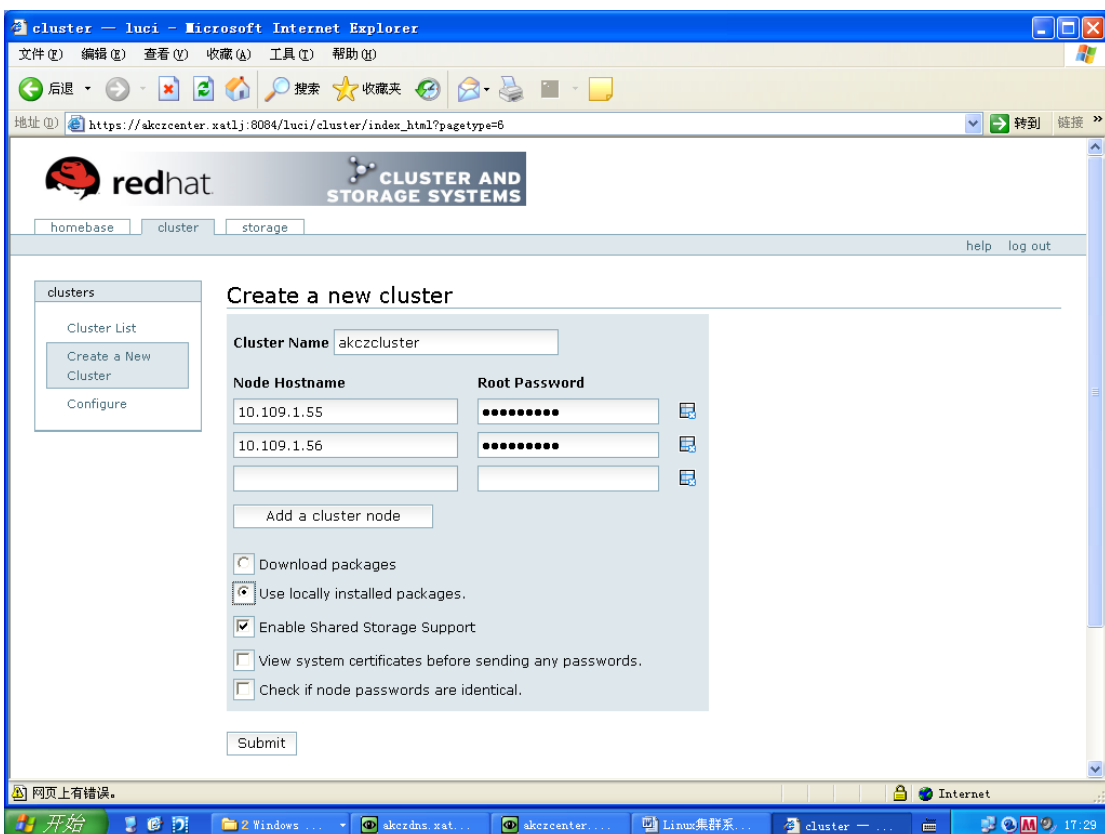


出现如下画面：

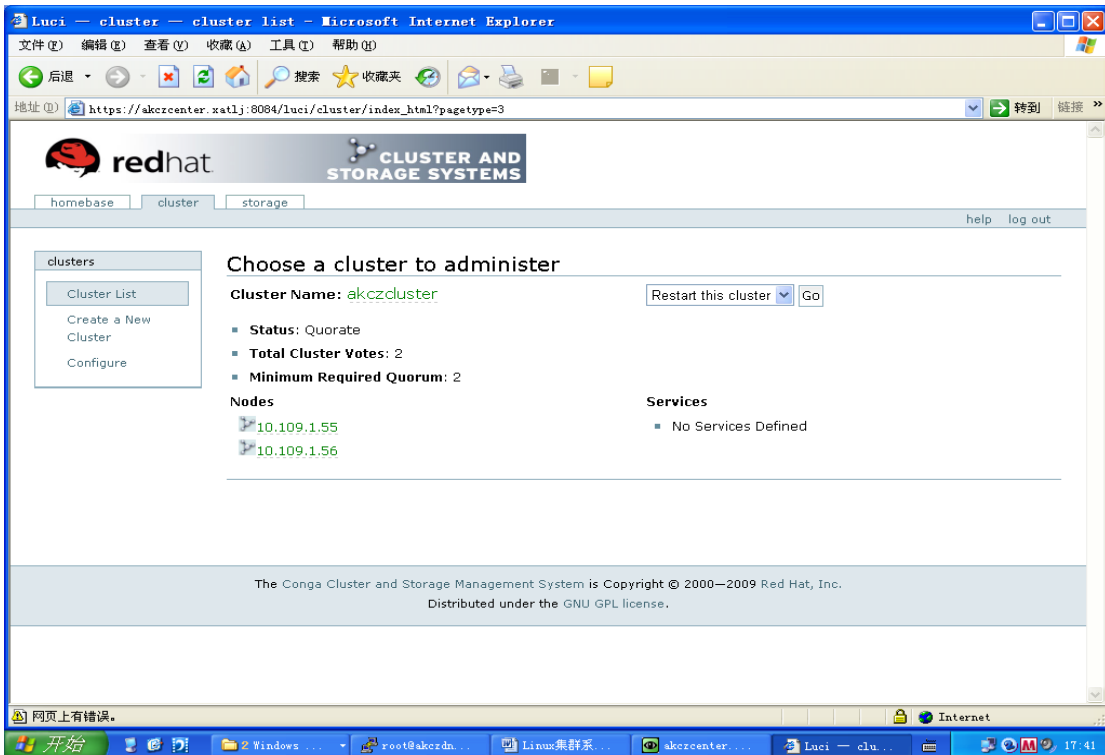


点击 cluster-

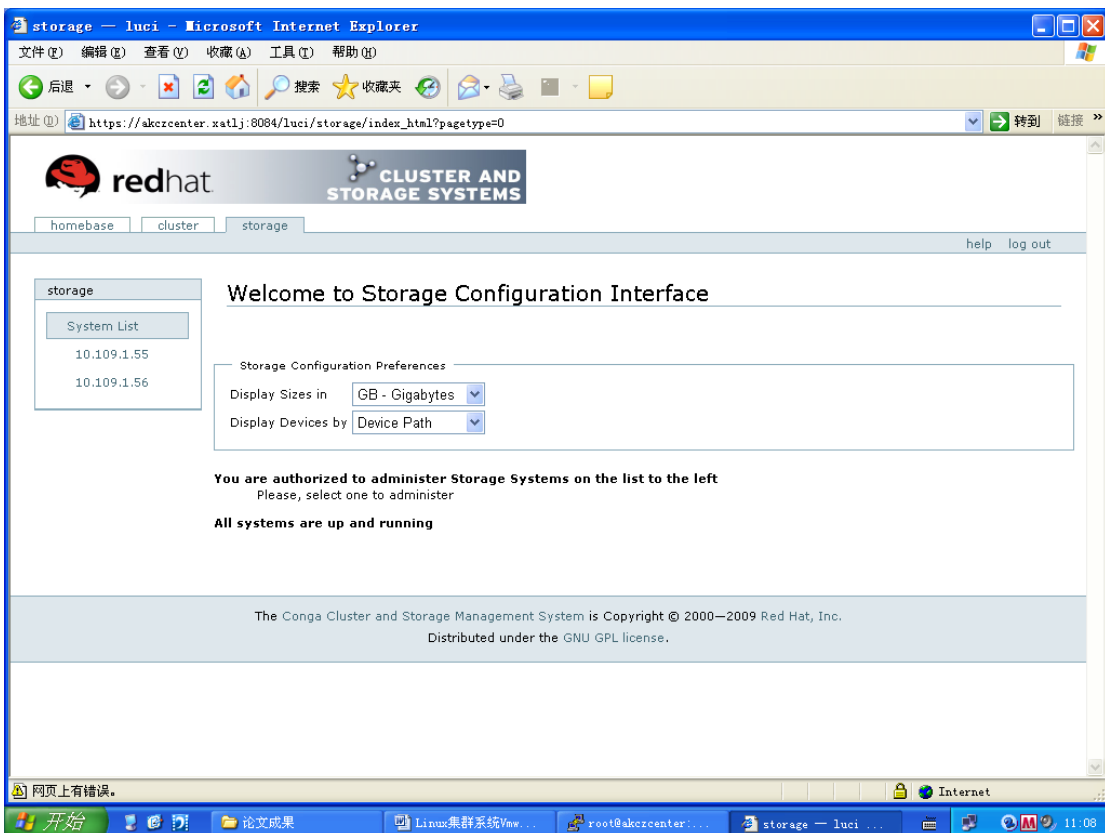
>create a new cluster 要求添入集群信息，输入之后出现如下画面：



点击 Submit,系统将创建 Cluster 并重新启动系统,重起后登陆点击 cluster 就可以看到如下画面:



点击 storage 出现如下画面:



分别点击 10.109.1.55、10.109.1.56 出现如下画面:

在 Mountpoint 和/etc/fstab Mountpoint 处添入/akczftpLV2 为挂载点，点击 apply。

之后分别登陆两个节点输入 mount 可以看到如下信息：

```
[root@akczcenter ~]# mount
/dev/rd/c0d0p2 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/rd/c0d0p1 on /boot type ext3 (rw)
/dev/rd/c0d0p5 on /vmware type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
none on /sys/kernel/config type configfs (rw)
/dev/sda on /akczftpLV2 type gfs2 (rw,hostdata=jid=0:id=262145:first=1)
```

说明/dev/sda 被顺利挂载。

输入 vi /etc/fstab 可以看到如下信息：

```
LABEL=/          /          ext3 defaults    1 1
LABEL=/boot      /boot      ext3 defaults    1 2
LABEL=/vmware    /vmware    ext3 defaults    1 2
tmpfs           /dev/shm   tmpfs defaults    0 0
devpts          /dev/pts   devpts gid=5,mode=620 0 0
sysfs           /sys       sysfs defaults    0 0
proc            /proc      proc defaults    0 0
LABEL=SWAP-rd/c0d0p3 swap        swap defaults    0 0
/dev/sda /akczftpLV2 gfs2 defaults 0 0
```

说明/etc/fstab 已经被更新。

我们在 10.109.1.55 下创建文件夹 aaa,登陆到 10.109.1.56 查看，如果文件夹同样存在，我们可以将其删除，然后返回 10.109.1.55 查看，文件夹消失，说明 DLM 锁方式 GFS2 文件系统也正常。

五、部署 VMWARE 虚拟机

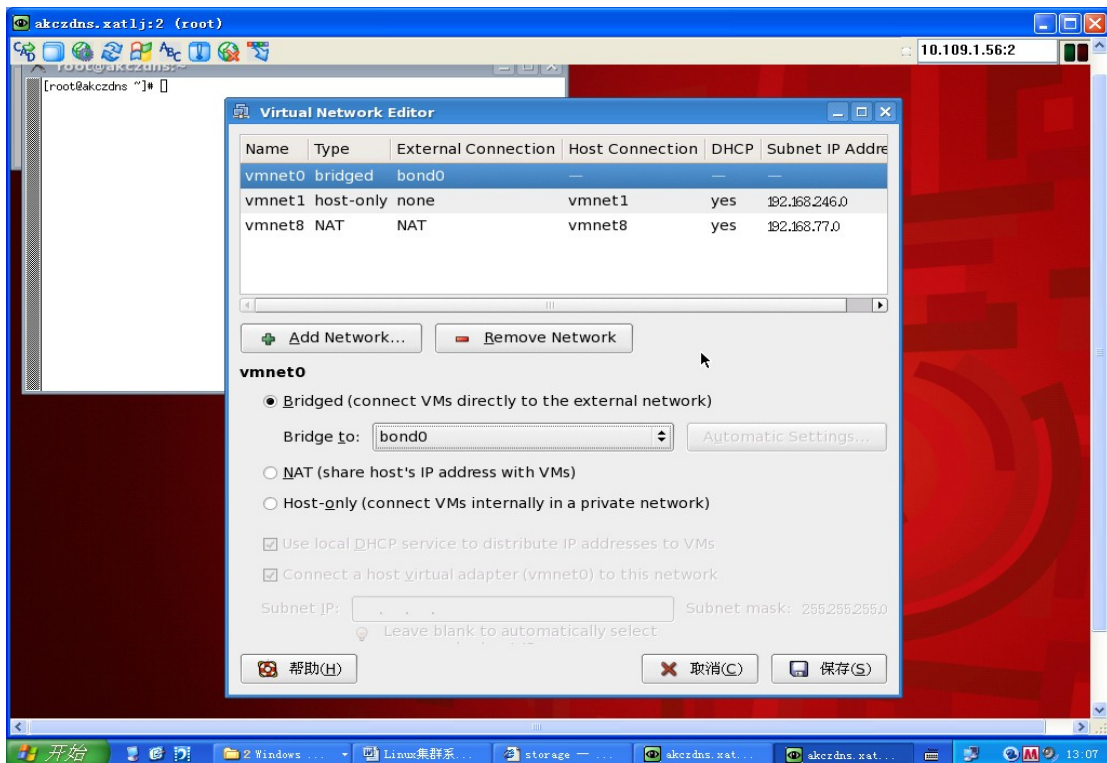
分别在节点中建立 vmware 帐号，分别在两个节点计算机中修改/akczftpLV2 目录的属组及用户为 vmware,确定修改成功后，查看如下图：

```

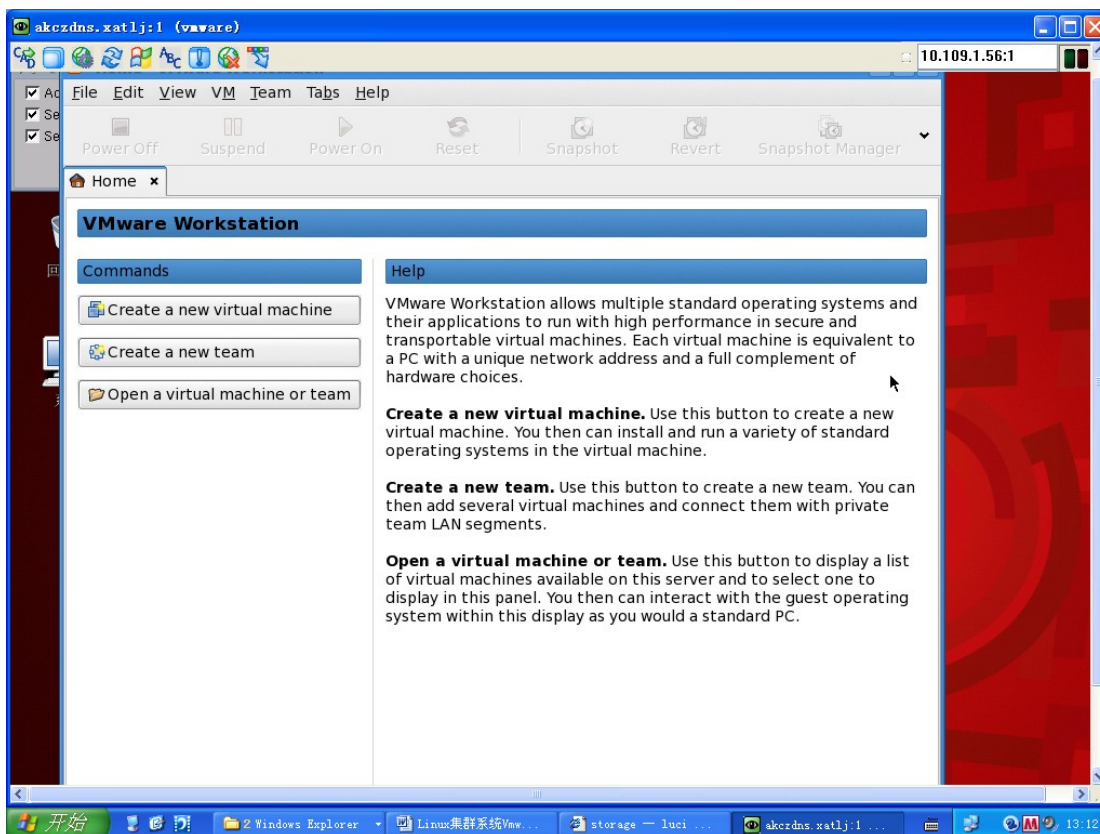
root@akczcenter:~ [125x43]
Connection Edit View Window Option Help
[root@akczcenter /]# drwxr-xr-x 2 root root 12288 2008-05-23 sbin
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxr-xr-x 2 root root 4096 2008-05-21 selinux
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxr-xr-x 2 root root 4096 2006-10-11 srv
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxr-xr-x 11 root root 0 05-19 01:32 sys
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxr-xr-x 2 root root 4096 2006-07-13 tftpboot
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxrwxrwt 12 root root 4096 05-19 11:15 tmp
-bash: drwxrwxrwt: command not found
[root@akczcenter /]# drwxr-xr-x 14 root root 4096 2008-05-21 usr
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# drwxr-xr-x 28 root root 4096 2008-05-21 var
-bash: drwxr-xr-x: command not found
[root@akczcenter /]# ls -l
-rwxr-xr-x 3 vmware vmware 3864 05-18 18:34 vmware
drwxr-xr-x 2 root root 4096 2008-05-23 bin
drwxr-xr-x 4 root root 1024 2008-05-21 boot
drwxr-xr-x 12 root root 3660 05-19 11:10 dev
drwxr-xr-x 128 root root 12288 05-19 13:01 etc
drwxr-xr-x 3 root root 4096 2008-05-22 home
drwxr-xr-x 14 root root 4096 2008-05-23 lib
drwx----- 2 root root 16384 2008-05-21 lost+found
drwxr-xr-x 2 root root 4096 2006-10-11 media
drwxr-xr-x 2 root root 4096 2007-09-25 misc
drwxr-xr-x 2 root root 4096 2006-10-11 mnt
drwxr-xr-x 2 root root 4096 2006-10-11 opt
dr-xr-xr-x 135 root root 0 05-19 01:32 proc
-rw-r--r-- 1 root root 342906880 2008-11-02 rhel-5.1-server_supplementary-1386-disc1.iso
drwxr-xr-x 22 root root 4096 05-14 04:40 root
drwxr-xr-x 2 root root 12288 2008-05-23 sbin
drwxr-xr-x 2 root root 4096 2008-05-21 selinux
drwxr-xr-x 2 root root 4096 2006-10-11 srv
drwxr-xr-x 11 root root 0 05-19 01:32 sys
drwxr-xr-x 2 root root 4096 2006-07-13 tftpboot
drwxrwxrwt 12 root root 4096 05-19 11:15 tmp
drwxr-xr-x 14 root root 4096 2008-05-21 usr
drwxr-xr-x 28 root root 4096 2008-05-21 var
drwxr-xr-x 6 vmware vmware 4096 04-07 09:59 vmware
[root@akczcenter /]#

```

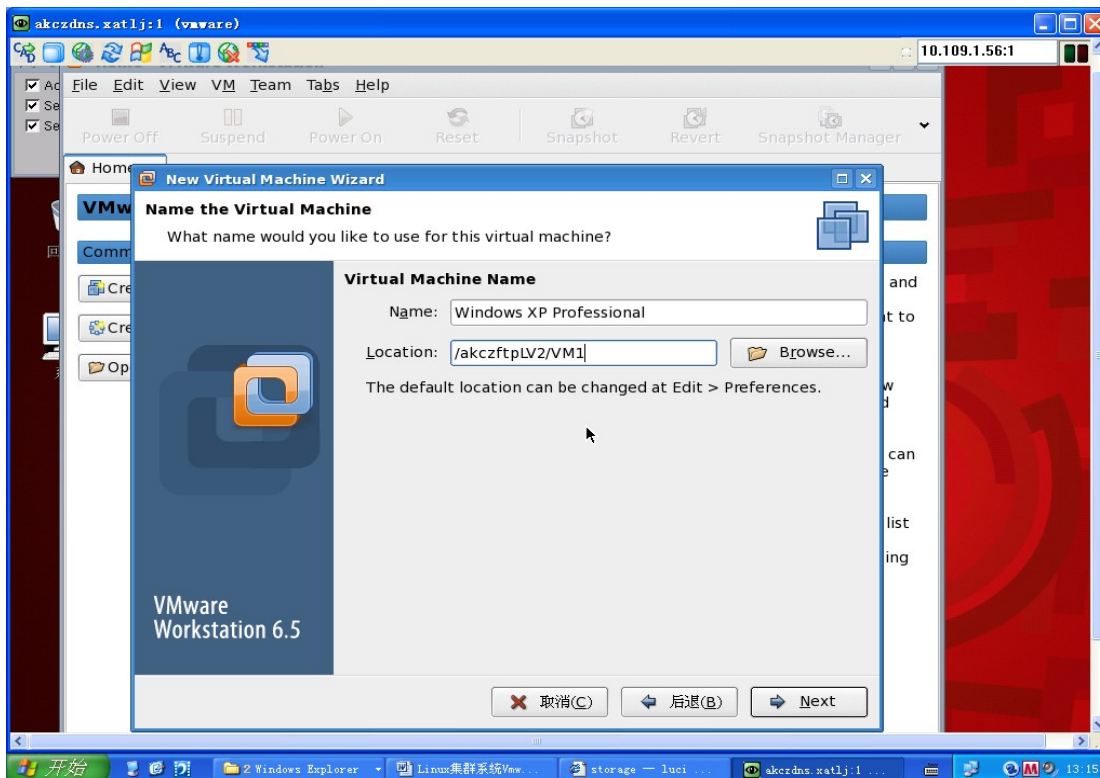
在节点一中安装 VMware-Workstation-6.5.2-156735.i386.bundle，并将 VMware 网络设置为桥模式，并将设备指向 bond0(节点均为双网卡绑定模式)。



使用 vmware 登入系统，点击 VMware Workstation 出现下图：

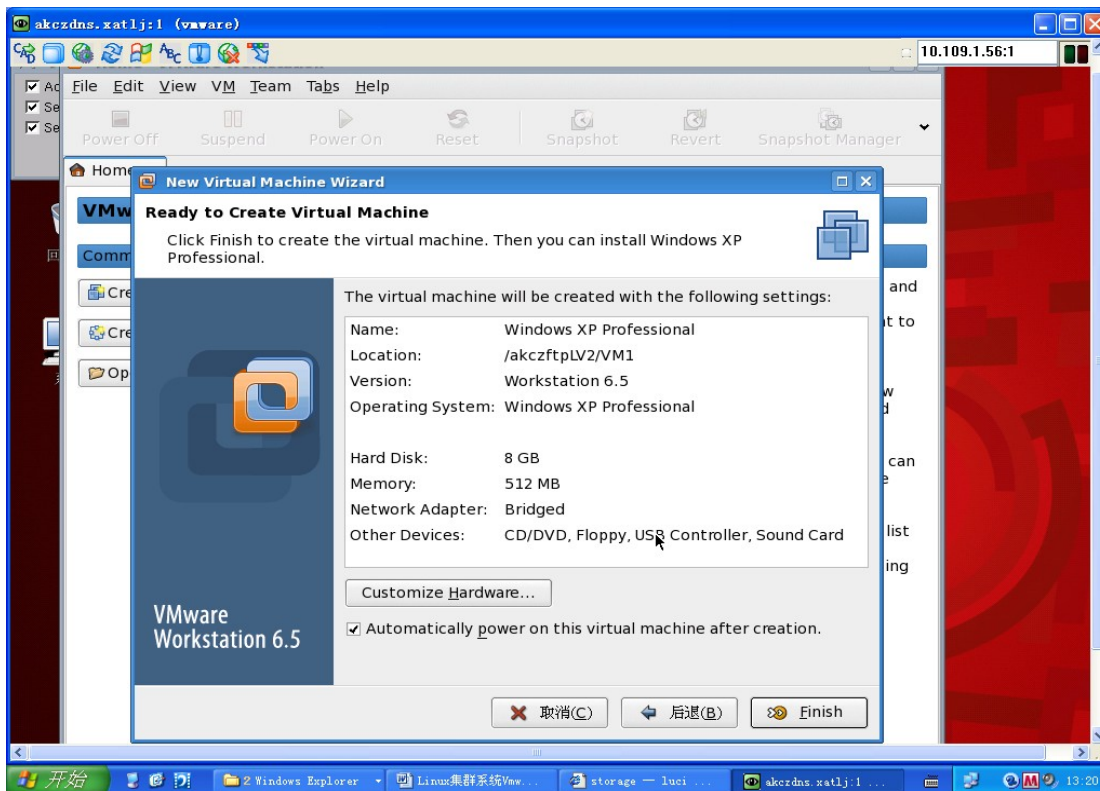


点击 Create a new virtual machine 创建虚拟机，到如下画面将 location 内容修改为/akczftpLV2/VM1



到如下画面将 File name 修改为/akczftpLV2/VM1 Windows XP Professional 点击 Next 出现如下

画面：



点击 Finish 完成虚拟机创建。

在节点二中同样进行配置。需要注意的是节点二中的 location 内容要修改为/akczftpLV2/VM2，File name 修改为/akczftpLV2/VM2 Windows XP Professional。

我们还可以在/akczftpLV2 上为 VM1 和 VM2 创建新的分区文件，部署 WINDOWS 系统应用。

六、系统的优点：

- 1) 使用了基于 iscsi 的 IP-SAN 技术，取代了价格昂贵的 SCSI 或者光纤 SAN 存储。
- 2) 应用系统和数据的有效分离，数据被集中存放在 RAID10 盘阵上，不会因为节点失效（软、硬件故障）导致数据丢失。
- 3) 虚拟机的快速切换应用。任意节点失效，VM 系统都可迅速在另一节点重新部署。
- 4) 应用分散，有利于系统均衡。10.109.1.59 既提供 FTP 服务，同时也是 ISCSI 服务器，10.109.1.55 作为 10.109.1.58 的宿主机，10.109.1.58 作为主 OA 应用服务器，日常访问频繁，为了减轻压力，10.109.1.55 只提供 VM 服务，10.109.1.56 作为 10.109.1.57 的宿主机同时也是 NTP 及 DNS 服务器，10.109.1.57 作为 WEB 服务器，其上部署了两个 WEB 网站。该应用方案目前为 150 个节点提供服务，没有出现过阻塞问题。
- 5) 使用了虚拟化技术，减少了硬件成本投入，提高了系统资源的使用率。

Linux-Netfilter&iptables 实现机制的分析及应用

Chinaunix 网友: Minit

一、引言

本文分析了 linux 下 netfilter/iptables 的实现机制，主要集中于分析用户态与内核态之间规则的关系，以及在此基础上对用户规则的解析的生成，然后分析了扩展模块的实现原理，并介绍了如何写一个扩展模块。本文分析的内核源码为 2.6.21.2，相应的 iptables 程序的版本为 1.3.7。由于自身水平有限，且相关的参考资料较少，不能保证文中的分析结论一定正确，如果读者发现问题，望不吝与作者联系。

二、用户态规则与内核态规则的关系

在 Netfilter/iptables 实现机制中，实际对数据包进行过滤的是内核态的规则，但为了用户能够对内核态的规则进行操作，需要将内核态的规则信息读取到用户空间，对用户空间的规则进行修改后，再根据用户态的规则信息设置内核态的规则信息。用户从内核态中获取规则信息的方式如图 2-1 所示，在该图中表示了保存规则的结构--表在用户态和内核态间的对应关系。

在内核态中，表 ipt_table 中的 private 成员指向的 xt_table_info 保存了该表所管理的规则。用户通过 getsockopt() 从内核态获取表的基本信息及规则内容，并将其保存在用户态中。

在用户态中，表 iptc_handle_t 保存了表管理的规则，在该结构中分别由 ipt_getinfo 保存表的基本信息，由 ipt_get_entries 保存表中的规则，在后面会对这些结构体进行详细介绍。为了便于用户对规则的操作，需要将表 iptc_handle_t 内的规则信息由函数 parse_table() 进行相应的处理，其会将规则按照链表形式分类存放，存放的格式如图 2-1 所示，有关用户态中链和规则的结构会在后面介绍。

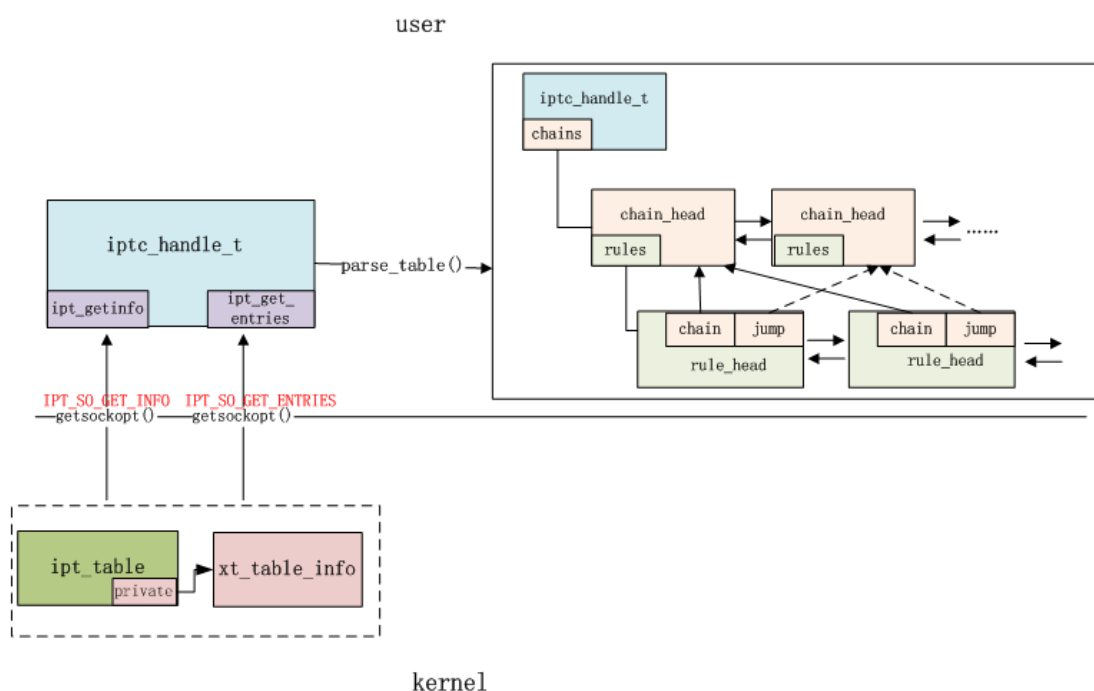


图 2-1 用户获取内核态规则信息

当从内核态将表的内容读取到用户空间后，用户就可以对表的内容进行修改，如在表内添加、删除、修改自定义子链等，在已有链内添加、删除、修改规则等。当对表的内容修改完后就对这些内容进行相应处理，然后通过 `setsockopt()` 更新内核态对应的表空间的内容。

为了方便将用户态的表内容更新到内核态，需要对表内的所有规则进行相应的处理，函数 `iptcc_compile_table_prep()` 主要计算表内链与规则对应的偏移值，然后将所有规则通过函数 `iptcc_compile_table()` 添加到结构体 `ipt_replace` 中。内核就是根据 `ipt_replace` 的内容来更新其表空间的内容的，前面计算的偏移值主要是为了明确规则所处的位置。用户操作内核态表内容的计数值是通过 `ipt_counters_info` 结构来进行通信的，内核根据从用户态传来的该结构体的值更新相应的内容。

用户定制的命令由函数 `do_command()` 处理，该函数会根据用户的需求修改表 `iptc_handle_t` 的内容，当对表的内容进行修改后，会同步更新内核态表 `ipt_table` 的内容，图 2-2 表示了用户设置内核态规则的流程。

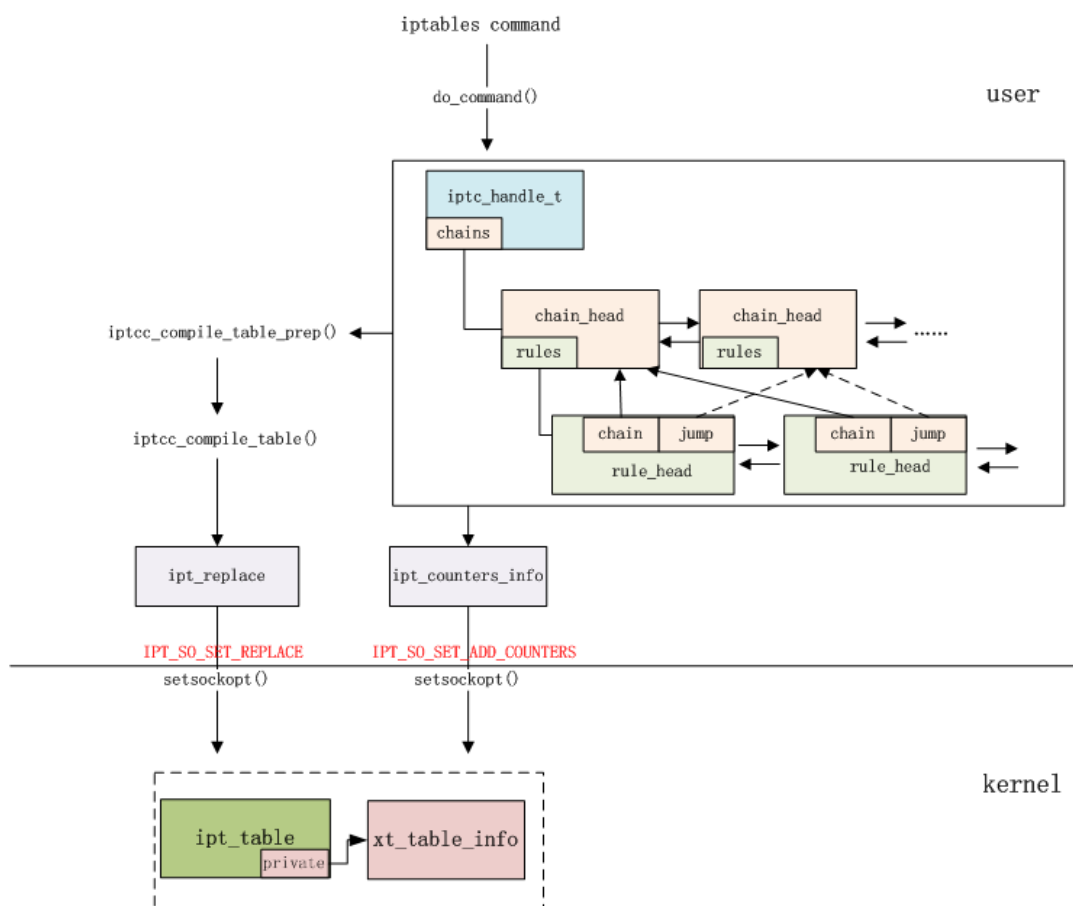


图 2-2 用户设置内核态规则信息

三、用户态规则的管理

1. 表的结构：

用户态中表结构 `iptc_handle_t` 如下图 3-1 所示，其中的 `info` 指向 `ipt_getinfo` 结构，`entries` 指向 `ipt_get_entries`，这两个结构体的内容都是内核态传递过来的，其中 `ipt_get_info` 保存了表的基本信息，`ipt_get_entries` 保存了该表所管理的规则信息，该结构中的 `entrytable` 所指向的内存位置保存了所有的规则 `ipt_entry`。当在用户态中对表内容进行处理后，会把规则内容组织到链中保存，其中 `chains` 指向了这些链。`chain_iterator_cur` 和 `rule_iterator_cur` 分别指向了当前遍历到的链与规则。

`iptc_handle_t` 结构的内容实质上与内核态中表结构 `ipt_table` 所表示的内容是一致的，只是其所存储的空间位置不一样。

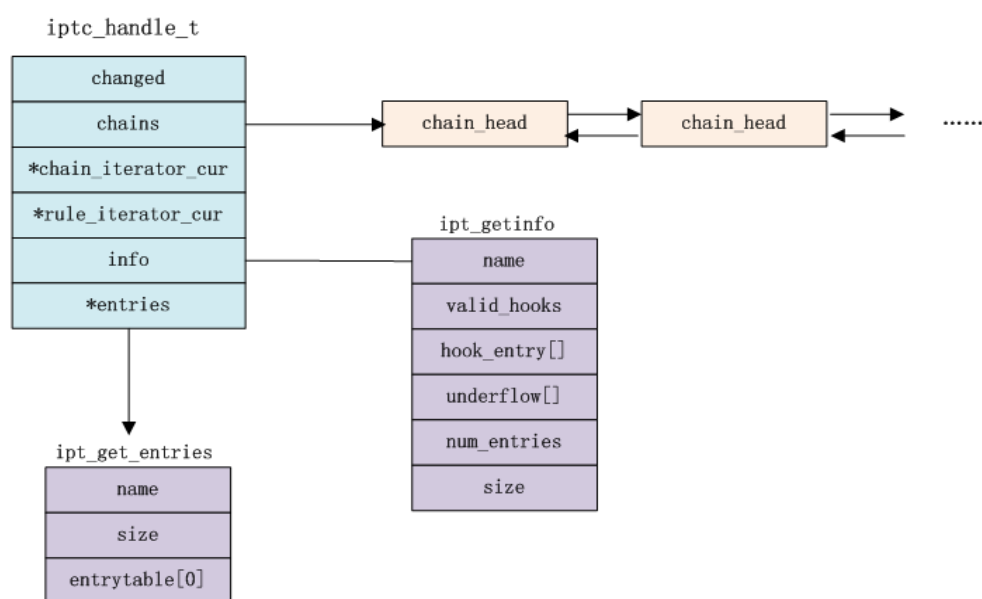


图 3-1 用户态中表的结构

1. 表中链的结构：

表 `iptc_handle_t` 中的 `chains` 指向该表所管理的所有链，每一个链用一个 `chain_head` 结构保存，该结构体的内容如下图 3-2 所示，有关该结构体内各个成员的具体作用在此就不做详细分析，其中 `counters` 记录了该链的统计值，`counter_map` 主要表示了对该链的统计值的操作，其在向内核态中设置相应的统计值时起作用。变量 `rules` 指向了该链所管理的所有规则，这些规则以链表的方式保存，一条规则用一个 `rule_head` 结构保存。

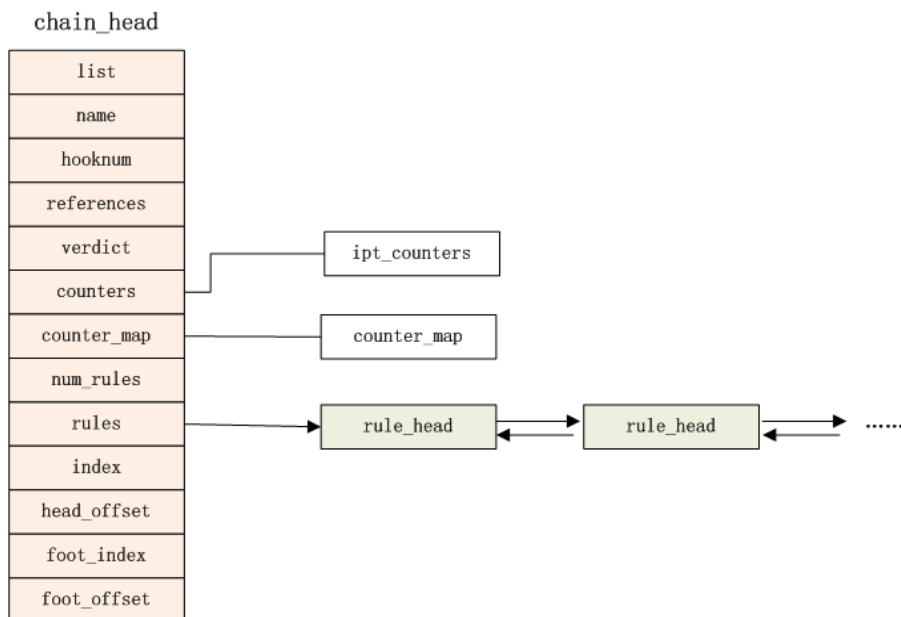


图 3-2 用户态中链的结构

1. 链中规则的结构及遍历：

规则 `rule_head` 的结构如下图 3-3 所示，在一条规则中，`chain` 指针指向该规则所属的链，而 `jump` 指针指向该规则将要跳向的其他链。`iptcc_rule_type` 表示了该规则所属的类型，如标准操作（ACCEPT、DROP 等）、扩展动作（SNAT 等）以及跳向其他链等等类型。为了方便内核态与用户态的信息交互，需要在交互前对表的内容进行处理，处理中就利用了链以及规则中的 `index`，`offset` 等变量值，有关处理的细节在此就不做详细分析。

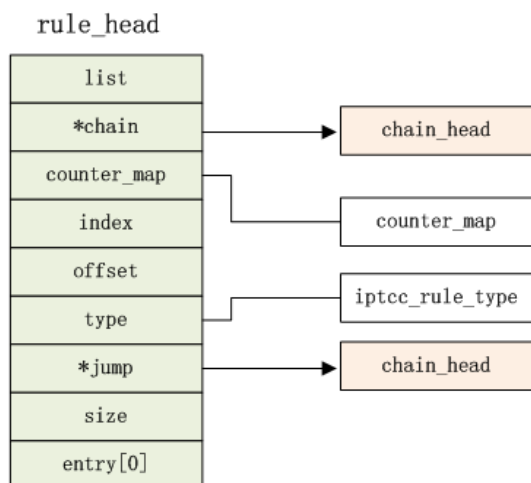


图 3-3 用户态中规则的结构

为了满足用户定制的策略，需要对表中的规则进行修改，这就需要实现对表中的规则的遍历，实现遍历的函数主要有：

`iptc_first_chain()`、`iptc_next_chain()`、`iptc_first_rule()`、`iptc_next_rule()`，通过这些函数可以找到表中的第一条链，当前链的下一条链，链中的第一条规则，当前规则的下一条规则。

四、用户态规则的解析

用户定制的命令实际上是对 iptables 指向的表里面的内容的操作，如下图 4-1 就表示了命令解析的流程，函数 generate_entry() 会按照用户制定的规则匹配条件及相应动作生成一条规则 rule_head，详细生成的流程会在下一节描述。在解析用户命令时，则根据用户指定的表、链及动作，对表的内容进行操作。如添加一条规则时，就会调用 iptc_append_entry() 将规则 rule_head 加到相应的链中去，iptables_delete_entry() 及 iptc_insert_entry() 则实现对规则的删除与插入。

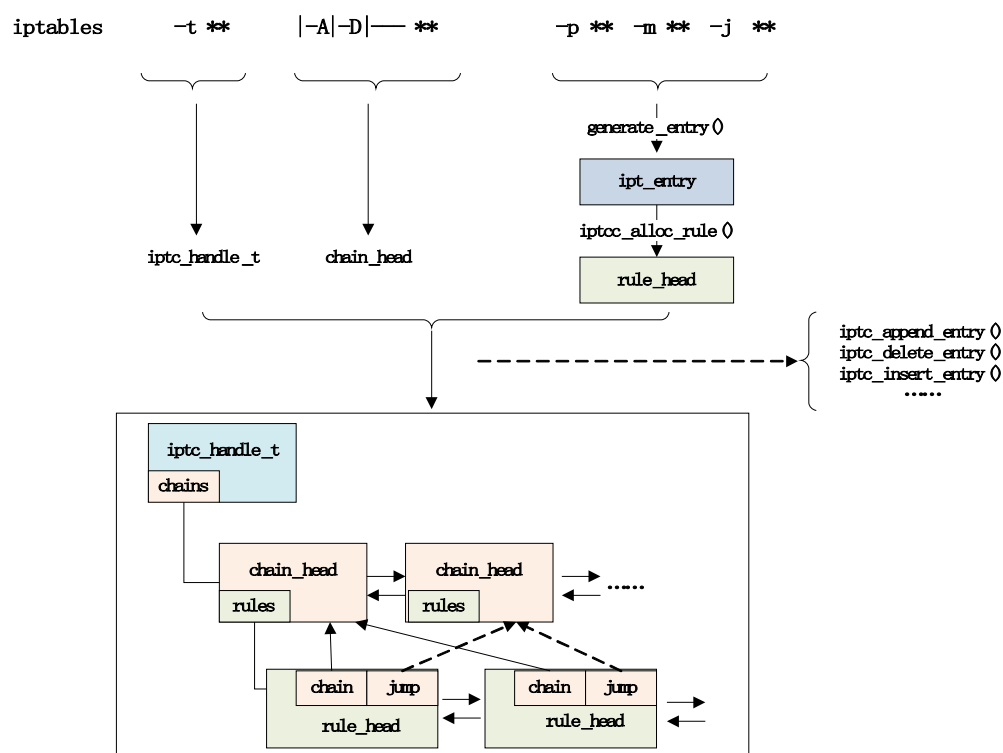


图 4-1 用户态规则的解析

五、用户态规则的生成

前面提到了在解析规则时，会将用户制定的匹配条件及动作转化为一条规则，转化的详细流程如下图 5-1 所示。从图中可以看出，用户态中规则的存储方式与内核中规则的存储方式是一样的，都是顺序存储。标准匹配条件部分会放在 ipt_entry 结构中，扩展匹配条件部分的转化相对较复杂，首先会调用函数 find_match() 在全局链表 iptables_matches 中查找相应的扩展模块，如果找到了就复制一个该扩展模块到链表 matches 中，matches 实际上就保存了这条规则中指定的扩展匹配模块，这两个链表在后面讲述扩展模块时还会详细介绍。然后调用函数 find_target() 设置该规则对应的动作，如果不是标准动作则需要查找全局链表 iptables_targets，以找到该动作对应的 iptables_target。

当前面已经把规则的不同部分作了处理后，下一步就通过函数 generate_entry() 将这些内容整合成一条规则，规则的开始部分标准匹配条件 ipt_entry，然后将 matches 指向的多个 iptables_match 中的 ipt_entry_match 结合在一起，形成规则的扩展匹配条件，最后根据前面处

理的动作部分设置 `ipt_entry_target`，形成规则的动作处理。动作处理中涉及到标准动作与扩展动作的不同处理，处理的具体细节在此不做详细分析。最后形成的规则如下图所示。

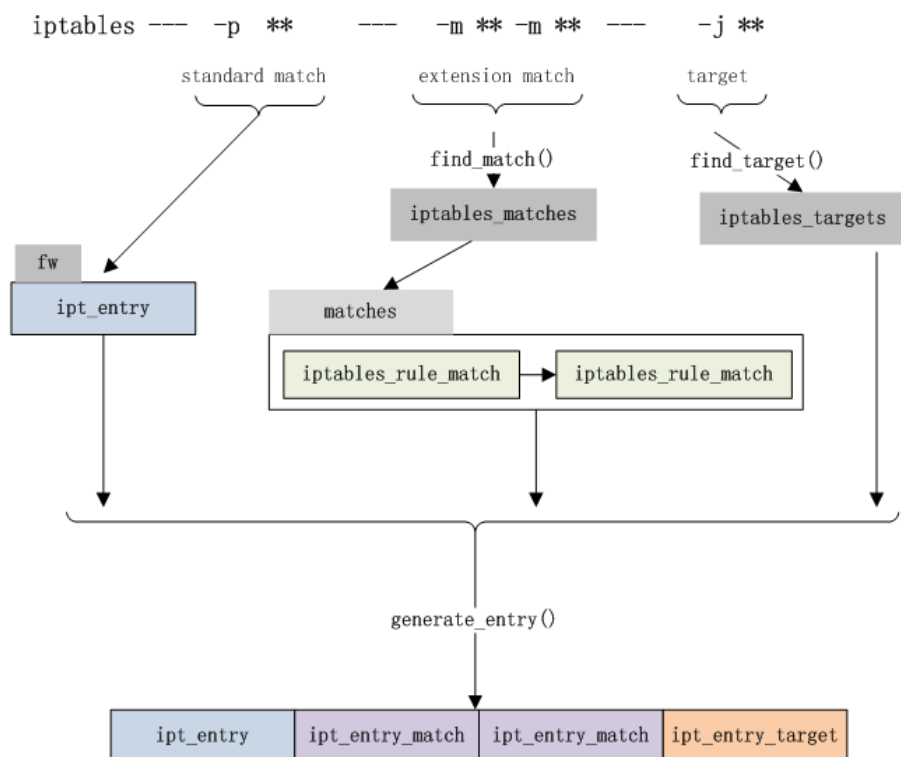


图 5-1 用户态规则的生成

六、扩展模块

1. 扩展匹配模块

扩展匹配模块的结构如图 6-1 所示，在该结构中有一个变量 `m` 指向匹配模块 `ipt_entry_match`，该结构与内核态中的结构是一样的。用户态中的扩展匹配模块的存储与管理方式与内核态是相似的，模块的注册调用函数 `register_match()` 实现，该函数将用户设定的 `iptables_match` 模块添加到全局变量 `iptables_matches` 指向的双向链表中，该链表如图 6-2 所示。

在扩展匹配模块中定义了一些函数，这些函数主要用于处理跟该模块相关的规则，如其中 `parse()` 主要处理与该模块相关规则的参数信息，该函数的使用在后面小节会涉及到，这些函数的具体细节，在此不做分析。

iptables_match

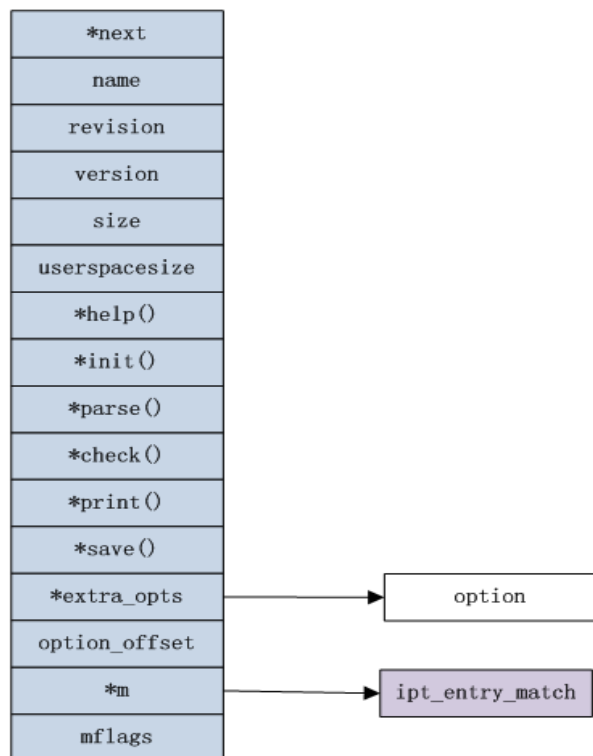


图 6-1 iptables_match 结构

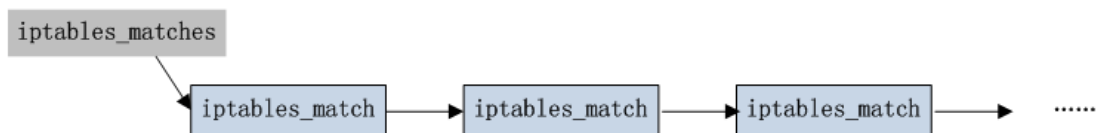


图 6-2 iptables_match 的存储

1. 扩展动作模块

扩展动作模块的结构如图 6-3 所示，该结构与扩展匹配模块的结构相似，其由全局变量 `iptables_targets` 指向的链表保存，链表的形式如图 6-4 所示。

iptables_target

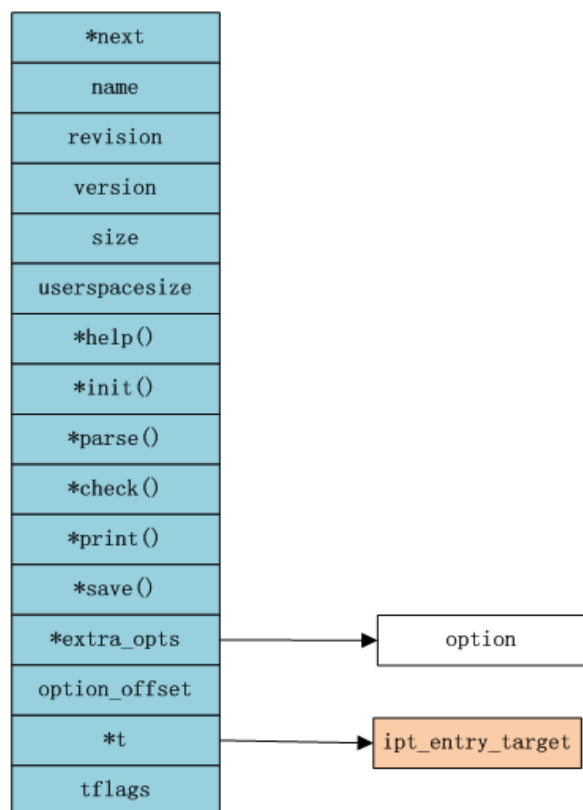


图 6-3 iptables_target 结构

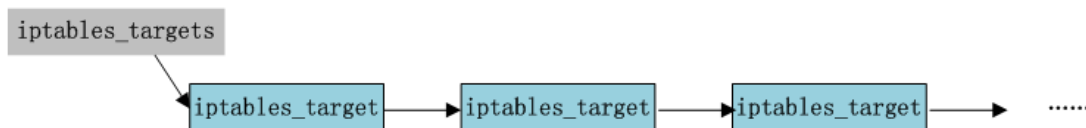


图 6-4 iptables_target 的存储

七、用户态与内核态之间参数的传递

在扩展模块中参数在用户态与内核态间的传递中，扩展模块的 `parse()` 会首先将用户定制的匹配参数放置在用户空间的结构体 `ipt_XX_info` (`XX` 表示扩展模块的名字，该命名法为通用的)，`ipt_XX_info` 的内存位置与扩展模块中 `ipt_entry_match` (`ipt_entry_target`) 的 `data` 内存位置相同，故这两者实际保存用户空间中相同的内容。

当把扩展模块 `ipt_entry_match` (`ipt_entry_target`) 的内容传到内核中后，提取内核中扩展模块的 `data` 内容，并将其强制转化为内核空间的 `ipt_XX_info` 类型的结构，这样的话，内核态的扩展模块就可以通过 `ipt_XX_info` 结构获得用户设置的参数，然后就可以调用该模块的处理函数如 `match()` (或 `target()`) 处理。图 7-1 表示了以扩展匹配模块为例的参数传递过程。

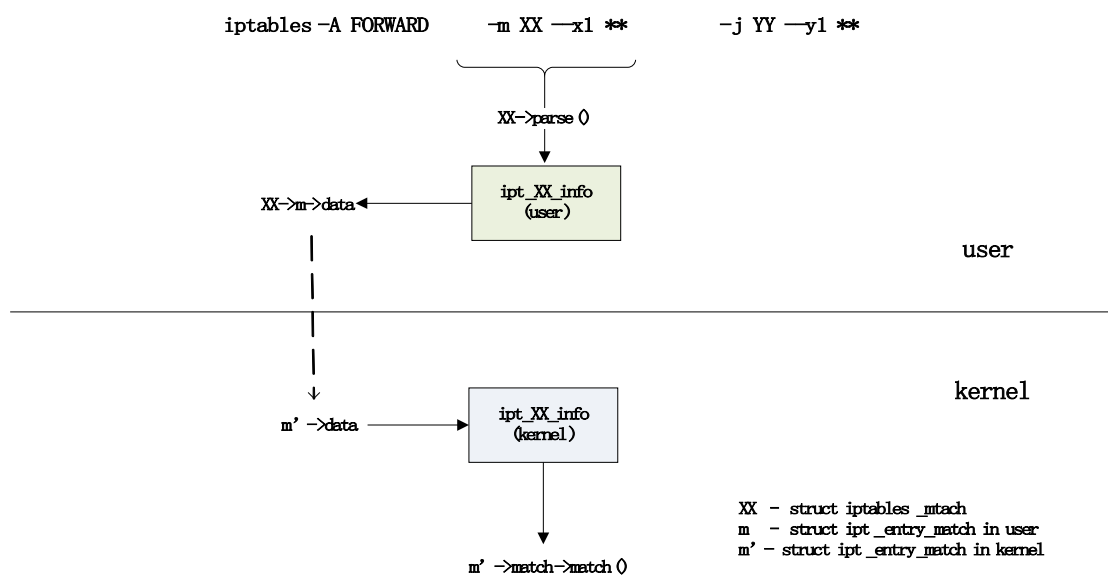


图 7-1 用户态向内核态传递参数

八、扩展模块的应用

通过前面的描述，可以看出扩展一个匹配模块需要做的工作。下面以扩展匹配模块为例，扩展动作模块的操作类似，主要包括以下内容：

用户空间：

- (1). 定义一个 iptables_match 模块并初始化相应的参数值，在模块的加载函数中调用 register_match() 将 iptables_match 模块注册到全局链表 iptables_matches 中；
- (2). 定义用户态与内核态共用的数据结构 ipt_XX_info，其主要用于交互数据；

内核空间：

- (1). 定义一个 ipt_match 模块并初始化相应的参数值，在模块的加载函数中调用 xt_register_match() 将 ipt_match 模块注册到 xt_af 的相应链表中；

当完成前面的工作后，就需要修改程序的编译文件及内核配置文件，以目前我分析的内核版本 2.6 为例，则需要修改如下文件：

- (1). 在 iptables/extensions/Makefile 文件的 PF_EXT_SLIB 中新加模块名 XX；
- (2). 在 linux/net/ipv4/netfilter/Makefile 中新加一行：obj-\$(CONFIG_IP_NF_MATCH_XX) += ipt_XX.o
- (3). 在 linux/net/ipv4/netfilter/Kconfig 中新加以下行：

```

config IP_NF_XX
tristate "....."
depends on .....

```

help

.....

To compile it as a module, choose M here. If unsure, say N.

完成以上步骤后，重新配置内核并选取该模块，编译后即可使用该扩展模块了。

九、总结

本文主要目的是分析 netfilter/iptables 实现机制的大框架，很多细节问题并未涉及到，希望通过此分析文档对进一步了解该机制提供帮助，并为使用该机制提供指导。由于自身水平有限，其中存在的问题望读者谅解并不吝指正。

从 WinCE 看 djyos 的生存空间和价值

ChinaUnix 网友: djyos

都江堰操作系统自发布半年来，一直得到嵌入式业界的关心和支持，经常收到热心朋友的 QQ、电话和邮件，他们给予了我极大的帮助和鼓励，无以为谢，我会继续努力，不使关心支持我的人失望。跟成都的戴先生认识，也是通过都江堰操作系统，某一天，戴先生在 QQ 上找到我。

戴：你现在 djyos 发展得怎么样了？

我：正在努力。

就这样开始我们的对话，通过 QQ 聊天，我知道戴先生是一个有 10 几年敬业的嵌入式业者，由于聊得比较投机，我们互相交换了电话。

前天下午，我的手机响起，是戴先生。

戴：嘿，我明天就飞深圳，可以见到你了。

我：太好了，到了给我电话。

.....

今天上午，在戴先生下榻的酒店见面了，瘦高个，很精干的一个小伙子。我们一边品着乌龙茶，一边聊天。聊到 wince 和 linux 和 djyos 的问题。

戴：作为投资者，他们关心的是，客户需要什么，并不关注你使用什么技术。

我：还有一点，他们也关注技术能给他解决什么问题。

戴：对，wince 和嵌入式 linux 的比较，linuxer 们津津乐道的是，嵌入式 linux 小巧，高速，功能强大，还免费；相较之下，同样的应用，wince 需要更多的存储器，更快的 cpu，以及价格不菲的 license 费用。从技术上讲，wince 能解决的问题，没有什么问题是 linux 解决不了的，反而可能存在一些嵌入式 linux 能解决而 wince 不能解决的问题。

我：然而，市场这个最权威最公正的裁判官青睐了谁呢？谁的装机发货量大呢？这些问题是不言而喻的。不客气地说，嵌入式 linux 现有的那点出货量，有不少是作为与微软谈判的筹码而存在的。为什么呢？

戴：一个操作系统要取得成功，必定具备某些优势，能解决其他系统解决不了的问题。我们知道

RTOS 为什么存在，是因为它解决了实时问题，wince 相对于嵌入式 linux，又解决了什么问题呢？而使人愿意花高价购买 license，并付出更高的硬件成本呢？有一部分是客户使用习惯的问题，但嵌入式应用中，用户使用习惯的惯性并没有 PC 中这么顽固。wince 成功在于，它解决了产品快速上市和研发成本的问题，这个研发成本主要是工程师的工资。有调查显示，从事嵌入式 linux 开发的工程师数量远比 wince 多，linux 工程师的工资也比 wince 工程师高，而更高的工资更多的工程师，所支撑的产品装机出货量，却……。难怪老板们会选择支付 license 而不要免费的嵌入式 linux 了。

我：既然 wince 的制胜之道如此，我们就没有理由怀疑 djyos 的前途，为什么呢？

我：嵌入式领域特别是在专业嵌入式领域的程序员，有许多是行业背景很深的专家，比如化学、生物学等。但往往缺乏计算机操作系统方面的知识，其他操作系统下，线程的出生入死皆由这些行业专家控制，而线程和进程知识，就是操作系统内核中的九阴真经，史上有几人懂九阴真经？在 djyos 下编程，则可以无需学习晦涩难懂的线程知识，甚至在多核环境中，依然不用。这无疑降低了人员培训费用和人工工资。

我：还有，djyos 以事件为调度核心，人们在编程中之需要弹出事件、编写事件处理函数，这就像在 RTOS 中引入了 VC 一般的开发环境，这无疑会大大加速开发进度，使产品能够快速上市。

再者，为项目经理贴心的各种功能设置，从系统的加载方式到 API 设置，都为项目经理组建和管理团队做了充分的考虑，使项目管理更容易、团队脉络更清晰、系统联调更轻松、后期维护更顺利，这也无形中降低了研发成本、加快了研发进度。

我们的对话结束了，但我并没有停止思考，来自 Gartner 公司的数据证实，自从 2003 年微软在嵌入式操作系统全球市场份额调查中荣登榜首以来，就一直保持这个成绩。另一方面，《电子产品世界》2008 年在工程师中做的调查却表明，使用 linux 应用开发的工程师是 wince 应用开发者的 2.5 倍。互联网上搜索一下 wince 和 linux 的技术指标对比，也是 linux 完胜。开发者众多，性能优越，市场占有率却不高，这是 linux 摆脱不了的尴尬。开发者众多还说明一个事实，linux 市场占有率低是因为 linux 应用产品的研发效率太低。我们再看看 wince 取胜的行业，往往是成本非常敏感的 PC 衍生产品上，这进一步说明研发效率的重要性。

wince 之所以取胜，在很大程度上来自快速方便的开发，快速推出产品，技术风险小。在 windows 系统衍生工具的支持下，事件触发式编程令人倍感方便。与 wince 通过应用程序（对操作系统来说，EVC 等开发工具属于应用程序）间接实现事件出发式编程相比，djyos 的事件触发确实原生的，即 djyos 调度器就是按照事件进行调度的。所以，你只要有一个支持 ansi C 的编译器，只要会 C 语言，再加一个文本编辑器，就可以实现事件触发式编程。原生式的事件触发编程，从理论上，其执行效率和代码尺寸也应该比用应用程序间接实现的要高。

以上是 djyos 的核心算法在通用嵌入式操作系统中的优势，此外，djyos 还是一个 RTOS，除跟 wince 在手持设备上竞争外，还可以与 VxWorks 等实时系统竞争，在 RTOS 领域，djyos 在研发方便性上的优势更加突出。在性能上，按 djyos “九九加一”原则设计的中断管理架构，使部分中断可以永不禁止状态，从而实现最快的中断响应延迟。我们知道，实时系统中，实时性要求最高的部分功能，往往是用中断实现的，最快的中断响应速度，意味着 djyos 具有最高的实时性。

有关更多的 djyos 操作系统讨论请访问 ChinaUnix 开源项目孵化专区：

<http://linux.chinaunix.net/bbs/forum-74-1.html>

GCC 开发计划及其分支介绍

ChinaUnix 网友: EricFisher

这篇文章大部分是对 GCC 网站上的原文进行的摘抄和翻译。其中, 开发计划部分, 参见 <http://gcc.gnu.org/develop.html>; 开发分支部分, 参见 <http://gcc.gnu.org/svn.html#devbranches>。另外还有一些信息来自 GCC summit 和 GCC mailing lists。

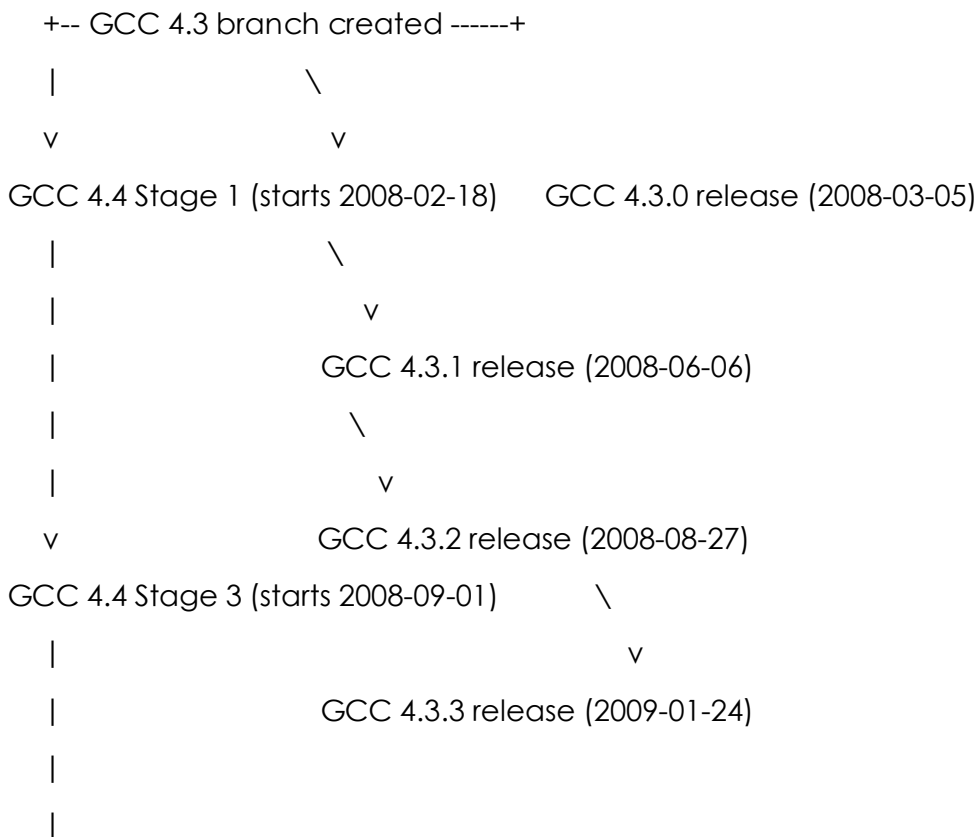
GCC 有一套完善的开发计划和方法。从而, 整个庞大的社区可以借助 SVN, mailing list, wiki 等工具, 进行有条不紊, 快捷高效的合作, 不断推出高质量的版本。

在开发主干 (trunk) 上, 开发过程需要经历三个阶段 (stage): 阶段一, 可以对编译器做任何自然的修改, 特别是从开发分支 (branch) 中合并进来的大的改动; 阶段二, 只允许有一些较小的改进; 阶段三, 主要是修复 bug, 另外如果一个新的 port 不需要改动编译器的其它部分, 则也可以加进来。

对于大的改动, 则要先创建一个开发分支, 在分支上进行开发, 当满足一定要求的时候, 才可以合并到开发主干上。这些要求包括: 一些基本的标准要求, 比如文档, 任何用户可见的改动, 包括命令行选项, 都要在手册中有介绍; 开发分支要至少在三个不同的目标机 (target) 上验证过, 包括编译器的自举和回退测试 (regression test)。

在阶段三快要结束的时候, 创建一个发行版本分支。发行版本分支主要用于修补相对之前版本有回退的地方, 从而使得每个发行版本都优于之前的版本。随后, 便是一些修补 bug 的版本。

整个流程可以使用如下的时间表来展示:



+-- GCC 4.4 branch created -----+

```
|          \  
v          v
```

GCC 4.5 Stage 1 (starts 2009-03-27) GCC 4.4.0 release (2009-04-21)

```
|  
|  
v
```

现在，GCC 4.4.0 已经发布，新合并进来的分支包括 ira, graphite 等。GCC 4.5.0 正处于第一个开发阶段，也有一些分支正在合并进来，包括 lto, plugins 等。这里列举了几个分支：

IRA (Integrated Register Allocator)

改进 GCC 的寄存器分配。使用集成的寄存器分配方法来替换以前分散在不同过程 (pass) 中的局部寄存器分配，全局寄存器分配等。寄存器分配问题在 GCC 中存在很久，一直没有得到很好的解决。现在这个分支已经合并到了 GCC 4.4.0 中。如果使用 -fdump-rtl-all-all 来转储

(dump) tree 级的中间过程，则可以看到 *.ira 文件。Vladimir 在 GCC Summit 2007 上有一篇文章《The integrated register allocator for GCC》，介绍了相关的工作。

Graphite (Gimple Represented as Polyhedra)

基于多面体模型在高层次上对循环进行优化。已经合并到 GCC 4.4.0 中。该分支用于进行新的改进，从而合并到以后的版本中。参见 <http://gcc.gnu.org/wiki/Graphite>。他们每周都有一次电话会议。Sebastian 在 GCC Summit 2006 上有一篇文章《GRAPHITE: Polyhedral Analyses and Optimizations for GCC》，介绍了相关的工作。并且，今年的 Google Summer of Code，也有一个相关的项目。

LTO (Link Time Optimization)

链接时优化，现在应该是由 Diego Novillo 维护（我不确定）。简单的讲，就是编译器先语法分析源程序，将中间表示保存在输出的目标文件中，在链接的时候，再次调用编译器将所有目标文件的中间表示都读进来一起优化。详细的介绍参见

<http://gcc.gnu.org/wiki/LinkTimeOptimization>。LTO 的设计文档写的很精炼概括，值得借鉴。

现在，这个分支正在向开发主干中合并。相信当 lto 合并进来之后，新版本的 GCC 在性能优化方面会有一个很好的提升。

Plugins

使得 GCC 具有支持插件的功能，一些新的优化和其它功能模块可以作为插件的形式提供给 GCC。在最新的 Internals 中，已经增加了 plugins 这一章节。参见 <http://gcc.gnu.org/wiki/plugins>。该分支已经合并到了开发主干中，并且已经关闭。

Gcc-in-cxx

在使用 c++ 重写了 binutils 中的链接器 (gold) 之后，Ian Lance Taylor 开始着手将 GCC 也改写成 C++ 的形式。GCC Summit 2008 中的《A New ELF Linker》介绍了相关的工作。gold 的代码，写得非常漂亮。Gcc-in-cxx 也应该非常值得期待。

Incremental-compiler

该分支用来将 GCC 改写成增量编译器 (incremental compiler)。增量编译器用来减少 GCC 的编译时间,使得编译时间大体上与源代码的改动大小成比例。该分支由 Tom Tromey 维护。在 GCC Summit 2008 上,有一篇《Incremental Compilation for GCC》介绍了相关的工作。

GCC 的 mailing list 每天都有很多的邮件,如果是关于一些开发分支的话题,一般会在标题上标明。最近有一些关于 LLVM 和 GCC 的话题相当激烈,包括 GCC 与 LLVM 的性能比较

(<http://gcc.gnu.org/ml/gcc/2009-05/msg00302.html>), 以及将 LLVM 作为 GCC 的插件

(<http://gcc.gnu.org/ml/gcc/2009-06/msg00055.html>)。社区中,有人认为在自由软件领域,有一些竞争是好事。这包括 LLVM 和其它编译器。就好比人们在开始一个新的项目时,也应该考虑一下使用什么语言更好。

网友热评

热点技术评论

[本人开发的 C 语言跨平台开发库](#)

[关于数组越界问题！](#)

[bash 判断语句问题](#)

[22 条经典的编程引言](#)

[求助：卷损伤之后的恢复](#)

[Linux/UNIX 下如何获得应用程序的版本号？](#)

[请教 fork\(\) 返回值的问题](#)

[程序结束时释放内存的必要性](#)

[做一个刷排名的程序，谁有好的方法？](#)

[刚用 C 写的 xml 解析代码](#)

[请教如何通过 socket 传 struct](#)

[用 UDP 建立可靠链接，你怎么做？](#)

[如何计算某 char 变量 bit 位为 1 的总数目？](#)

[linux 清空文件夹命令有吗](#)

[关于 Nagios 辅助软件](#)

[4 核的机器能装 redhat linux 5 吗？](#)

[TX QQ for Linux 什么时候有更新啊！](#)

[抉择 Ubuntu 还是 Fedora-谁是真正的 Linux](#)

[Nand ECC 校验纠错原理及内核 ECC 代码分析](#)

[新手应该如何阅读别人的代码？](#)

[硬链接是什么意思？](#)

[SWAP 启用问题](#)

[磁盘/空间不足问题](#)

[给新手的一份 LVM 实验笔记](#)

[如何破解 ROOT 密码？](#)

热点新闻评论

[再见，Linux 计算机！](#)

[一个做 IT 苦力三年的人的生活现状](#)

[中国为什么少有自学成才的高手](#)

[传腾讯上千台服务器遭受攻击 搜狗嫌疑最大](#)

[龙芯：本土 CPU 家族的“理想主义色彩”](#)

[绿坝称未偷美软件公司代码 只是有类似之处](#)

[php 程序员 3 年后的困惑](#)

[eclipse 我是在受不了你了](#)

[PERL 语言，我看行！](#)

[甲骨文再出损招 OpenOffice 阵营面临瓦解](#)

[有没有 linux 方面的项目？](#)

[HP 笔试归来，HP 你不得不对咱鄙视你](#)

[Sun 取消 Rock 高端芯片开发 数十亿美元打水漂](#)

[华为面试题解法，麻烦大侠们看看思路是否正确](#)

[龙芯无奈购美公司专利授权 “中国芯” 光环褪尽](#)

[腾讯公司后台服务器经典面试题](#)

[胡伟武关于购买 MIPS 结构授权的几点说明](#)

[WM 大势将去？欧洲运营商倒戈 Android](#)

[中国何时才能从开源消费者成为开源贡献者](#)

[龙芯 -- 中国 CPU 核自主战略并非失败](#)

[中国计算机专业失业者最多](#)

[过了 RHCE 能找到一份网管的工作吗？](#)

[中国统治 NSA 支持的 TopCoder 竞赛](#)

[昨天,今天,明天! Unix 系统的 40 年](#)